

## ABSTRACT

Title of dissertation: Using Domain-Specific  
Information in Image Processing

Brianna R. Cash, Doctor of Philosophy, 2014

Dissertation directed by: Professor Dianne O’Leary  
Department of Computer Science  
& Institute for Advanced Computer Studies

With the increasing availability of high resolution imaging tools, even in our pockets (i.e. smartphones), everyday users can do far more than simply digitally capturing a family moment. The ease of new applications available in these portable forms, linked with users who have expert knowledge about the images and tasks, opens the door to new possibilities. With this in mind we propose two new approaches that utilize the user’s knowledge for improved results. We apply these approaches to real life problems in medical and scientific image applications.

In the first approach, we introduce a class of linear and nonlinear methods which we call Domain-Specific Grayscale (DSGS) methods. A DSGS method transforms a color image into an image analogous to a grayscale image, where user-specified information is used to optimize a specified image processing task and reduce the computational complexity. We introduce new methods based on projection into the space of single-coordinate images, and we adapt support vector machines by using their scores to create a DSGS image. We apply these methods to applications

in dermatology, analyzing images of skin tests and skin lesions, and demonstrate their usefulness.

In the second approach, we introduce a tool for improved image deblurring that safeguards against bias that can easily be introduced by a user favoring a particular result. This is particularly important in scientific and medical applications used for discovery or diagnosis. We provide real-time results of choices of regularization methods and parameter selection, and we check the statistical plausibility of the results, using three statistical diagnostics, allowing a user to see the results of the choices.

Our work demonstrates the utility of domain-specific information, supplied by the user, in improving the results of image processing algorithms.



Using Domain-Specific  
Information in Image Processing  
Applied Mathematics Scientific Computing (AMSC)

by

Brianna R. Cash

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2014

Advisory Committee:  
Professor Dianne O'Leary, Chair/Advisor  
Professor Radu Balan  
Professor David W. Jacobs  
Professor Wolfgang Losert  
Professor Laurent Montesi

Dedicated to my husband, Jason

## Acknowledgments

There are many people who deserve acknowledgement for helping me get to the completion of my thesis, which at times seemed like an impossible task.

First I would like to thank my advisor and mentor Dr. Dianne O’Leary for providing such amazing guidance and support. She gave me opportunity to explore the questions and ideas that I found most interesting while giving me invaluable guidance on how to approach those ideas. She provided opportunities to attend summer schools and conferences which both gave motivation and perspective.

To my committee, Dr. Radu Balan, Dr. David W. Jacobs, Dr. Wolfgang Losert, and Dr. Laurent Montesi, they deserve thanks for agreeing to being on my committee and for providing great feedback and thought provoking questions.

I would like to thank the AMSC program and staff, in particular Alverda McCoy and Dr. Konstantina Trivisa, for support throughout my entire time at the University of Maryland.

This opportunity to pursue and complete my Ph.D. would not have been possible without my husband’s countless sacrifices and patience. I thank him for both believing in me and supporting me in pursuing my dreams.

I would like to thank my parents for their unconditional love, support and encouragement over the years and for being such amazing role models.

I would like to thank my officemate and friend Victoria Taroudaki for keeping me up-to-date on deadlines so that graduation was a possibility and for putting up

with me for the five years we shared an office, always being supportive as we moved through the journey of grad school together.

I would like to acknowledge Dr. Herman Mitchell, Agustin Calatroni, and Jeremy Wildfire of Rho Inc. and 2012 SAMSI Industrial Mathematical & Statistical Modeling Workshop for Graduate Students for granting me access to the skin-prick test images collected as well as encouragement to continue the work on the problem they presented. I would also like to acknowledge Ian McColl, FACD for granting permission to use the photos from the Atlas of Dermoscopy.

This work was made possible by the financial support of the NSF Grant DMS 1016266.

Lastly I would like to thank my son for being my motivation to finish; he did not make it easy to finish but he made it worth it.

# Table of Contents

List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Part I: Domain-Specific Grayscale For Improved Image Processing . .	2
1.2 Part II: A Tool for Graphical Image Deblurring Exploration (GIDE) .	5
1.3 Part III: Conclusion and Future Work . . . . .	6
I Domain Specific Grayscale	7
2 Linear Domain-Specific Grayscale Transformations for Color Images	8
2.1 Background . . . . .	10
2.1.1 Digital Images . . . . .	10
2.1.2 Standard Representations for Color Images . . . . .	11
2.1.3 Representation of Grayscale Images . . . . .	13
2.1.4 Other Representation of Color Images . . . . .	13
2.1.4.1 Cylindrical Coordinate Color Spaces . . . . .	14
2.1.4.2 Opponent Color Spaces . . . . .	15
2.1.5 Modified Color Spaces for Specific Applications . . . . .	16
2.1.5.1 Skin Color Models . . . . .	16
2.1.5.2 Pipe Image Interpretation Based on Support Vector Machines . . . . .	17
2.2 Previous Work in Color Edge Detection . . . . .	17
2.2.1 Decomposed Edge Detection . . . . .	18
2.2.2 Vector Based Edge Detection . . . . .	18
2.3 Projection to Find New Domain-Specific Grayscale Color Space . . .	19
2.3.1 Computing the Projection . . . . .	22
2.3.1.1 Maximizing Distance between Pixels (MDP) . . . . .	24
2.3.1.2 Minimizing Area between Distributions (MAD) . . .	24
2.4 Use of Support Vector Machines to Determine a New Color Space . .	26

2.4.1	Hard Margin SVM . . . . .	26
2.4.2	Soft Margin SVM . . . . .	28
2.5	Thresholding Images for Segmentation . . . . .	29
2.6	Comparing the Linear Domain-Specific Grayscale Methods . . . . .	31
2.6.1	Natural Images . . . . .	31
2.6.1.1	Berkeley Segmentation Dataset . . . . .	31
2.6.1.2	Synthetic Data . . . . .	32
2.6.1.3	Implementation of Linear DSGS Methods . . . . .	34
2.6.2	Results on Synthetic Data . . . . .	35
2.6.3	Results on Natural Images . . . . .	40
2.7	Summary and Conclusions . . . . .	43
3	Nonlinear Domain-Specific Grayscale Transformations for Color Images . . . . .	44
3.1	Kernel SVM . . . . .	45
3.2	Characteristics of Kernels . . . . .	50
3.2.1	Constructing Kernels . . . . .	51
3.2.1.1	Polynomial kernels . . . . .	52
3.2.1.2	Gaussian Kernels . . . . .	52
3.2.2	Conditionally Positive Definite Kernels . . . . .	53
3.2.2.1	Log Kernel . . . . .	54
3.2.3	Other Notable Kernels . . . . .	54
3.3	Kernel and Kernel Parameter Selection . . . . .	55
3.3.1	Selecting a Kernel . . . . .	55
3.3.2	Parameter Selection . . . . .	57
3.4	Kernel SVM as a Domain-Specific Grayscale Method . . . . .	57
3.5	Non-Kernel Nonlinear Domain-Specific Grayscale Method . . . . .	58
3.6	Results with the Non-Linear Domain-Specific Grayscale Methods . . . . .	61
3.6.1	Synthetic Image Testing . . . . .	62
3.6.2	Results on Natural Images . . . . .	67
3.7	Advantages and Disadvantages of Kernel SVM DSGS Methods . . . . .	68
3.8	Summary and Conclusion . . . . .	72
4	Using Domain-Specific Grayscale Method for Dermatology Digital Image Analysis . . . . .	74
4.1	Imaging-Aided Diagnosis in Dermatology . . . . .	75
4.1.1	Preprocessing Images . . . . .	76
4.1.2	Multilayered Skin Model of Absorbance . . . . .	77
4.1.3	Boundary Detection . . . . .	79
4.2	Measuring Skin Erythema and Wheal Size . . . . .	79
4.2.1	Previous Work in Detecting and Measuring Erythema and Wheals . . . . .	81
4.2.2	Motivation for Domain-Specific Grayscale Method . . . . .	83
4.2.3	Dataset . . . . .	84
4.2.4	Summary of Results of Measuring Erythema . . . . .	85
4.2.5	Summary of Results of Measuring Wheal . . . . .	88

4.3	Border Detection of Skin Lesions . . . . .	102
4.3.1	Past Work in Skin Lesion Boundary Detection . . . . .	104
4.3.2	Motivation for Domain-Specific Grayscale Method . . . . .	105
4.3.3	Dataset . . . . .	105
4.3.4	Summary of Results on Border Detection of Skin lesions . . . . .	106
4.4	Summary and Conclusion . . . . .	113
II	User Aided Parameter Selection for Image Deblurring . . . . .	121
5	GIDE: A Tool for Graphical Image Deblurring Exploration . . . . .	122
5.1	Background . . . . .	123
5.2	Mathematical Model of Regularization Methods . . . . .	125
5.2.1	Continuous Representation . . . . .	128
5.2.2	Discrete Representation . . . . .	128
5.2.3	Constructing the Blurring Matrix from the Point Spread Function . . . . .	129
5.3	Regularization Methods . . . . .	129
5.3.1	SVD-Based Regularization Methods . . . . .	130
5.3.2	TV Regularization for the Continuous Problem . . . . .	131
5.3.2.1	Formulation of Newton's Method . . . . .	132
5.3.2.2	Linearization Based on Introducing a New Variable . . . . .	132
5.3.3	Discretization of the TV Regularization Method . . . . .	133
5.3.3.1	Discrete Formulation of Newton's Method . . . . .	134
5.3.3.2	Formulation of Primal-Dual Newton's Method . . . . .	134
5.3.3.3	Relationship to the Dual Problem . . . . .	135
5.3.3.4	Implementing TV Regularization Method . . . . .	137
5.4	Initial Parameter Selection . . . . .	139
5.4.1	Generalized Cross-Validation (GCV) . . . . .	139
5.4.2	Discrepancy Principle . . . . .	140
5.5	Statistical Diagnostics . . . . .	140
5.5.1	Choice of Diagnostics . . . . .	141
5.5.1.1	Residual Diagnostic 1 . . . . .	141
5.5.1.2	Residual Diagnostic 2 . . . . .	142
5.5.1.3	Residual Diagnostic 3 . . . . .	142
5.5.2	Validation of Residual Diagnostics . . . . .	143
5.6	The Software . . . . .	144
5.7	Results and Testing . . . . .	145
5.7.1	Test Images . . . . .	145
5.7.2	Signal-to-Noise Ratio Effect on Diagnostics . . . . .	146
5.7.3	Effects of $\gamma$ on Computation Time . . . . .	146
5.7.4	Results on Larger Images and Varied PSF . . . . .	148
5.8	Summary and Conclusions . . . . .	150

III	Conclusions and Future Work	151
6	Conclusions and Future Work	152
6.1	Part I: Domain-Specific Grayscale For Improved Image Processing . .	153
6.1.1	Key Contribution . . . . .	154
6.1.2	Future Work . . . . .	155
6.2	Part II: A Tool for Graphical Image Deblurring Exploration (GIDE) .	157
6.2.1	Key Contribution . . . . .	157
6.2.2	Future Work . . . . .	158
A	Duality	159
A.1	Nonlinear Duality . . . . .	159
A.2	Duality for SVM . . . . .	161
A.3	Duality for TV . . . . .	162
B	Level Set Methods	164
B.1	Level Set Methods for Boundary Detection . . . . .	165
	Bibliography	167



## List of Tables

2.1	Percent misclassified for the synthetic images. . . . .	40
2.2	Percent misclassified for each image for a subset of Berkeley Segmentation Database. . . . .	41
3.1	Percent misclassified for the MAD method and the nonlinear methods on the synthetic images. . . . .	64
3.2	Percent misclassified for the MAD method and the nonlinear methods on the Berkeley Segmentation image dataset. . . . .	68
3.3	Percent misclassified and training time for the MAD method and nonlinear SVM methods for the image <i>Lions</i> . . . . .	69
4.1	<b>Fitzpatrick Phototype Scale:</b> This scale was developed in 1975 as a scale to measure the a patients skin phototype (sensitivity to UV radiation) and is the primary scale used today. This scale is often used by dermatologist as a scale of skin type/color [69]. . . . .	82
4.2	Table of preprocessing methods for detecting and measuring erythema and wheals. See Section 4.2.4 and Section 4.2.5 for more details. . . .	84
4.3	Relative $\Delta EI$ for each method of the erythema images. . . . .	89
4.4	Time to compute DSGS for the erythema images in seconds. . . . .	89
4.5	Time to compute DSGS methods for the wheal images in seconds. . . .	96
4.6	Table of preprocessing methods for border detection of skin lesions. See Section 4.3.1 for more details. . . . .	105
4.7	Time to compute DSGS methods for the Dermatoscope images. . . .	109
4.8	Time to compute DSGS methods for the standard lesion images. . . .	109
5.1	For 1000 runs, number of times the Diagnostics fail to be satisfied. $I(i) = 1$ if $(i - 1) \bmod(100) = 0$ or $(i - 2) \bmod(100) = 0$ and $I(i) = 0$ otherwise. . . . .	144

## List of Figures

1.1	Image of the skin after a skin prick test. The goal is to identify the reddening area that develops after the skin is irritated. Source: Rho Inc. . . . .	3
2.1	Left: Color image. Center: Grayscale image of color image. Right: Edge map of grayscale image after using the Canny edge detection method. . . . .	9
2.2	A color image broken into its red, green, and blue coordinates, along with the grayscale version. . . . .	12
2.3	A color image of M&Ms (left) and the grayscale image (right) computed using MATLAB'S <code>rgb2gray.m</code> . Source: <a href="http://blogs.mathworks.com/images/steve/2010/mms.jpg">http://blogs.mathworks.com/images/steve/2010/mms.jpg</a> . . . . .	14
2.4	Schematic of the <b>HSV</b> color space. Source: Image created by (3ucky(3all using Borland Delphi 2006, Adobe Photoshop 9.0. Source: <a href="http://upload.wikimedia.org/wikipedia/commons/e/e0/HSV_cylinder.png">http://upload.wikimedia.org/wikipedia/commons/e/e0/HSV_cylinder.png</a> . . . . .	14
2.5	Schematic of the 1976 CIELab color space. . . . .	15
2.6	Left: Synthetic image created in MATLAB with two contrasting colors that appear to be the same color in the <b>GS</b> image. Right: <b>GS</b> image found using <code>rgb2gray.m</code> . The blue and red boxes mark the user selected background ( $\bar{\mathbf{P}}$ ) and foreground ( $\mathbf{P}$ ). . . . .	20
2.7	Scatter plot of the pixels in the background ( $\bar{\mathbf{P}}$ , blue) and the pixels in the foreground ( $\mathbf{P}$ , red). . . . .	21
2.8	Scatter plot of the pixels in the background ( $\bar{\mathbf{P}}$ , blue) and the pixels in the foreground ( $\mathbf{P}$ , red) after rotation. . . . .	21
2.9	Resulting image after rotating the image and projecting onto the coordinate $\tilde{\mathbf{R}}$ from Figure 2.6. . . . .	22
2.10	Image of a patient's skin after an allergy test is administered. Source: Rho Inc. . . . .	23
2.11	The area in gray is the area over which we are minimizing. . . . .	25

2.12	Schematic of hard margin SVM where the two classes of data are represented by the filled and empty dots respectively. Source: <a href="http://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png">http://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png</a> .	28
2.13	Subset of images from the Berkeley Segmentation Database.	32
2.14	MATLAB's grayscale images for the Berkeley Segmentation Database images.	33
2.15	Human-user-filled edges for Berkeley Segmentation Database images.	33
2.16	Synthetic data created from real images.	34
2.17	MATLAB's grayscale images for synthetic data created from real images.	35
2.18	Resulting images after applying the different linear DSGS methods on the 'Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ in red and blue respectively.	36
2.19	Resulting images after applying the different linear DSGS methods on the 'Lions 2 Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ in red and blue respectively.	37
2.20	Resulting images after applying the different linear DSGS methods on the 'Pepper Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ in red and blue respectively.	38
2.21	Resulting images after applying the different linear DSGS methods on the 'Goat Dot' image along with the histogram over all the pixel values with the pixels of the user selected $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ in red and blue respectively.	39
2.22	Comparing the <b>GS</b> image to the results of MAD and SVM for the images of lions and image of horses.	41
2.23	Image comparing the <b>GS</b> image to the results of MAD and SVM for the image of an owl, a tiger and a goat.	42
3.1	Image of a Tiger from the Berkeley Segmentation dataset with user selected background and foreground for training.	45
3.2	Selected background (blue box) and foreground (red box) pixels from Figure 3.1 plotted in <b>RGB</b> space.	46
3.3	Left: Two dimensional toy problem where the background (red box) is a donut around the ball of foreground (blue box). Right: After mapping to the three dimensional feature space where there exists a hyperplane that separates the two data types.	47
3.4	Level sets for two-dimensional pixels. Contour plot of $\Psi(\mathbf{x}, u)$ where the red stars are the foreground pixel and blue dots are the background. The dark blue curve is the zero level set.	61
3.5	Results of the nonlinear DSGS methods on the "Pepper Dot" image. In the histograms on the right, pixels of $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ are shown in red and blue respectively.	65

3.6	Results of the nonlinear DSGS methods on the “ <i>Owl Dot</i> ” image. In the histograms on the right, pixels of $\mathbf{P}_w$ and $\bar{\mathbf{P}}_w$ are shown in red and blue respectively. . . . .	66
3.7	Resulting DSGS image for MAD, gSVM, pSVM and lSVM for the image of an Owl with the human-determined boundary superimposed in red. The images are displayed so that values less than zero are set to black to highlight the classification of the foreground. . . . .	69
3.8	Results for two different selected backgrounds of significantly different sizes. The images are displayed so that values less than zero are set to black to highlight the classification of the foreground. . . . .	70
3.9	Color image of M&Ms and resulting DSGS images for the MAD, gSVM, GS, pSVM and lSVM. . . . .	72
4.1	Image of skin after prick test on four different participants of different phototypes. Source: Rho Inc. . . . .	82
4.2	Six images used for measuring erythema after prick test. Source: Rho Inc. . . . .	85
4.3	Six images used for measuring wheal after prick test. Source: Rho Inc. . . . .	86
4.4	W4(left) and W5 (right) with hand-drawn boundary of wheal. Source: Rho Inc. where the boundaries were drawn by Anna Gleason. . . . .	86
4.5	E1: Comparing the choice of training background. Displayed such that pixels less then zero are displayed as black to emphasis on the classification of pixels. . . . .	89
4.6	E1: GS, Log,a of Lab, MAD, and lSVM. . . . .	90
4.7	E2: GS, Log,a of Lab, MAD, and lSVM. . . . .	90
4.8	E3: GS, Log,a of Lab, MAD, and lSVM. . . . .	91
4.9	E4: GS, Log,a of Lab, MAD, and lSVM. . . . .	91
4.10	E5: GS, Log,a of Lab, MAD, and lSVM. . . . .	92
4.11	E6: GS, Log,a of Lab, MAD, and lSVM. . . . .	92
4.12	Scatter plot of pixels of $\mathbf{P}$ (red) and $\bar{\mathbf{P}}$ (blue) in <b>RGB</b> space for images E1-E5. . . . .	93
4.13	E2, E4, and E5. Resulting boundary after Chan-Vese to Single-Coordinate Image. Starting contour (white), lSVM (red), MAD (green), “a” of Lab (black), and GS (blue). . . . .	94
4.14	W1: GS, Log,a of Lab, MAD, and lSVM. . . . .	97
4.15	W2: GS, Log,a of Lab, MAD, and lSVM. . . . .	97
4.16	W3: GS, Log,a of Lab, MAD, and lSVM. . . . .	98
4.17	W4: GS, Log,a of Lab, MAD, and lSVM. . . . .	98
4.18	W5: GS, Log,a of Lab, MAD, and lSVM. . . . .	99
4.19	W6: GS, Log,a of Lab, MAD, and lSVM. . . . .	99
4.20	Scatter plot of pixels of $\mathbf{P}$ (red) and $\bar{\mathbf{P}}$ (blue) in <b>RGB</b> space for images W1-W6. . . . .	100
4.21	Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), lSVM (red), MAD (green), and GS (blue) for W1- W6. . . . .	101

4.22	Image comparing the human drawn edge to the resulting boundary of the DSGS image for W4 (left) and W5 (right). Hand-drawn (black), LSVM (red), and MAD (green).	102
4.23	Skin Lesions, variety of colors. First two are moles, second two are cancerous. Source: First two images are from the Dermoscopy Atlas. Second two images are from DERMOFIT.	103
4.24	Dermatoscope images of a variety of skin lesions. Source: Dermoscopy Atlas.	107
4.25	Standard photo taken on a digital camera. Source: Top left from Dermoscopy Atlas. Remaining from DERMOFIT.	108
4.26	D1- Top: Color, GS, -LogR. Bottom: "B" or RGB, "b" of Lab, MAD, and LSVM.	110
4.27	D2- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	110
4.28	D3- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	111
4.29	D4- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	111
4.30	D5- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	112
4.31	D6- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	112
4.32	D7- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	113
4.33	Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), and GS (blue) for D1- D7.	114
4.34	S1- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	115
4.35	S2- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	115
4.36	S3- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	116
4.37	S4- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	116
4.38	S5- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	117
4.39	S6- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	117
4.40	S7- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	118
4.41	S8- Top: Color, GS, -LogR. Bottom: "B" of RGB, "b" of Lab, MAD, and LSVM.	118

4.42	Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), and GS (blue) for S1- S8. . . . .	119
5.1	Left: Blurred Image, Center: Deblurred Image, Right: True Image which has “train tracks”. Without knowledge of the true image having “train tracks” one might accept the deblurred to be a good image without realizing that important information was lost in the process. Source: Dianne O’Leary. . . . .	123
5.2	Restorations of $256 \times 256$ blurred satellite image provided in <b>RestoreTools</b> with zero boundary conditions, SNR=9 and parameters chosen by the initial parameter methods detailed in Section 5.4. . . . .	126
5.3	Final implementation of the GUI with the three diagnostics . . . . .	127
5.4	Results of Newton-CG for Total Variation regularization for two images. Top is a $16 \times 16$ image generated in MATLAB and the bottom image is $64 \times 64$ Modified Shepp-Logan generated in MATLAB. . . . .	137
5.5	For the $16 \times 16$ segment of the image “ <i>cell.tif</i> ” the range ( $\pm 0.0025$ ) was found for Diagnostic 1 using Tikhonov regularization. From the plot we can see that as the Signal-to-Noise Ratio decreases the range of parameters satisfying the diagnostic increases. . . . .	147
5.6	The difference in $\log_{10}$ of computational time (seconds) for the TV regularization method for parameters between $\gamma = 1$ and $\gamma = 10^{-9}$ . For all image sizes the maximum time was for $\gamma = 1$ . . . . .	148
5.7	$129 \times 129$ image of “ <i>cell.tif</i> ” with Gaussian blur and zero boundary conditions with SNR of 60. . . . .	149

## Chapter 1: Introduction

Today most people have access to digital cameras or smart phones. These imaging devices are often high resolution and do not require time to develop. With this availability users are finding new uses for this information. There are numerous applications on the market that help users utilize the camera to do a number of tasks beyond capturing an image.

We list a few applications that utilize the camera or images taken on the smart phone for a number of interesting tasks. The smartphone application Leafsnap identifies tree species from a user captured image of a leaf [44, 46]. The application EasyMeasure allows users to measure the distance of an object [24]. There are numerous applications that attempt to deblur an image taken on the smart phone. There are also a number of applications that aid in medical care delivery that utilize the camera including an application called Colorimetrix (in development) which can be used for monitoring conditions such as diabetes, kidney disease, and urinary tract infections [18, 93] and the Handyscope (recently approved by the FDA) that, with the addition of a lens, turns a smart phone into a Dermatoscope which is used for monitoring and diagnosis of skin lesions such as malignant skin cancer [36].

One advantage of these applications is that the user, who has the most in-

formation about the image or task, has full access to the data. This additional knowledge can be essential to image processing. In our work we develop methods that enhance processing by utilizing often essential user information.

In Part I we introduce methods that can transform a color image into a domain-specific grayscale (DSGS) image, making tasks such as edge detection or segmentation easier and more computationally feasible. In Part II we develop a prototype tool to enable practitioners to deblur an image by enabling them to choose an effective regularization method for their particular problem. Even if these practitioners have minimal knowledge of the challenges in solving ill-posed problems, our method safeguards against bias that may be introduced by preconceived notions of what the visual results should be.

## 1.1 Part I: Domain-Specific Grayscale For Improved Image Processing

In Part I, we use user-specified color information to transform a color image into a single coordinate image. This transformation aids in detecting features that may not be initially evident while reducing the redundancy of information. For the task of finding edges in an image, there are widely used edge detection methods in the literature [53]. These methods are often based on measuring discontinuities in the images and are fast and easy to implement when applied to grayscale images. Unfortunately grayscale images only capture the light intensity of the continuous scene, losing information about the color (chroma). In some applications, this loss



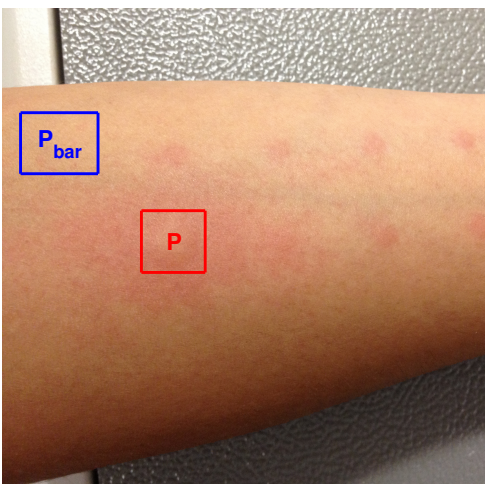


Figure 1.1: Image of the skin after a skin prick test. The goal is to identify the reddening area that develops after the skin is irritated. Source: Rho Inc.

of information is unacceptable. At the same time color images are expensive to store and process, since they involve three color coordinates (often Red, Green, and Blue (**RGB**)) instead of the single coordinate (i.e. grayscale images).

This work was inspired by the problem presented by Rho Inc. as part of the SAMSI 2012 Industrial Math/Stat Modeling Workshop for Graduate Students [64]. Their problem involved finding the boundaries of a wheal (raised hive) and erythema (reddening) that develops after positive skin-prick tests as seen in Figure 1.1. In this application the boundaries can be irregular and faint and traditional methods for finding the edges have limited success. We were granted access to the data used for the workshop.

In Chapter 2 we overview past work in finding color transformation of traditional **RGB** color images, discuss past work of image processing tasks on color images and introduce a new class of linear domain-specific grayscale (DSGS) trans-

formations. We do this by utilizing user-specified information about the foreground and background to find the orientation in the **RGB** color space that best differentiates between the patches. We present three different DSGS methods. The first method finds the orientation that maximizes the distance between the background and foreground pixels. The second method finds the orientation that minimizes the overlapping area between the distribution of the background and foreground pixels if we assume they are normally distributed. The final method uses a support vector machine (SVM) to find the orientation that separates the regions with maximum margin.

In Chapter 3 we extend the idea of finding a domain-specific grayscale to include nonlinear transformations. A nonlinear transformation is needed, for example in the cases where the background and foreground is made up of more than two distinct colors and there exists no hyperplane that separates the background and foreground in **RGB** color space. We introduce two nonlinear DSGS methods. The first method learns the nonlinear relationships between data points by assuming that the points are heat sources and heat sinks and finds the level set that best separates them. The second method takes advantage of well established kernel methods, which implicitly find a higher dimension space where the background and foreground pixels are linearly related. We apply the kernel methods to the dual formulation of the SVM.

In Chapter 4 we apply the DSGS methods to real life applications in dermatology. In practice, visual inspection as well as visual imaging is essential to diagnosing different skin conditions including skin cancer [51]. That being said,

there is variation between experienced and inexperienced dermatologists regarding to the success rate for diagnosing skin cancer [51]. Therefore, noninvasive digital visualization techniques and computerized diagnoses are becoming essential tools in care delivery [6]. In this chapter we apply our techniques for finding a DSGS image to aid in finding the boundary in two common dermatological tasks. Detecting erythema and wheals after skin-prick tests which are used by dermatologists and allergists to diagnosis allergies, is the first task we tackle. The second task involves finding the boundaries of skin lesions, which is a first step in computerized diagnostic systems for the diagnosis of skin cancer and other serious skin conditions. These two cases are challenging for traditional image processing tools as there is significant variation inherent to skin, and the boundaries can be faint and irregular. We show that use of DSGS methods can improve the results of standard image processing tools such as edge detection methods, traditionally developed for grayscale images, in dermatological applications.

## 1.2 Part II: A Tool for Graphical Image Deblurring Exploration (GIDE)

In Part II of our work we develop a methodology and software with a graphical user interface (GUI) that can be used by practitioners to choose an appropriate regularization. We call this software package Graphical Image Deblurring Exploration (GIDE). This package gives practitioners the ability to compare regularization methods while limiting the bias they may introduce by finding a solution that matches

what they expect, missing what is unexpected. This is done by guiding their choice of method and parameter by showing solutions and the statistical diagnostics for each choice in real time.

In Chapter 5 we review the components that make up the user-guided **GIDE** package for improved image deblurring. We review how to find the numerical solution to the truncated singular value decomposition, Tikhonov and Total Variation regularized problems, the automated parameter selection methods used for finding the initial guess for the regularization parameter for a given regularization method, and the statistical diagnostics [72]. We also give an overview of the development of the GUI and demonstrate how the GUI can be used by practitioners with little or no knowledge of the numerical methods.

### 1.3 Part III: Conclusion and Future Work

In Chapter 6 we discuss the impact of the work and discuss some future work related to domain-specific information in image processing.

## Part I

### Domain Specific Grayscale

## Chapter 2: Linear Domain-Specific Grayscale Transformations for Color Images

There are many accepted and well-used edge detection and segmentation methods that are fast and easy to implement when applied to grayscale images. Unfortunately grayscale images only capture the light intensity of the continuous scene, losing information about the colors (chroma). The literature states that 90% of the information in color images is captured in the grayscale image [61], but there are foreseeable applications where that additional 10% is critical in finding the relevant edges, as demonstrated by Figure 2.1. Grayscale images are measured in a single coordinate, where color images are measured in a three-coordinates space making any processing task much more costly.

In practice we are usually interested in a subset of edges in an image, such as edges defining a person or object. We can use the knowledge of the object or area of interest within the image as additional information to improve the results of edge detection. This work develops a new color space through user-aided transformation of the three-coordinate space into a single coordinate space where the color discontinuities of interest are captured. We call this new class of methods Domain-Specific Grayscale method or DSGS methods.

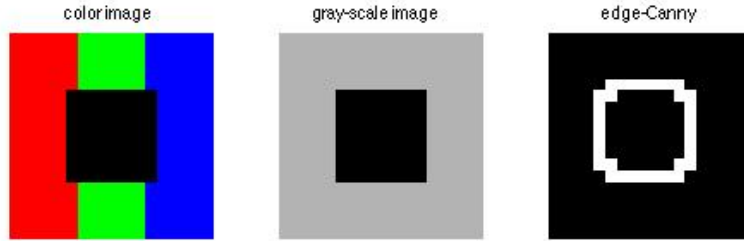


Figure 2.1: Left: Color image. Center: Grayscale image of color image. Right: Edge map of grayscale image after using the Canny edge detection method.

Applications of this work include enhancing images of skin for improved diagnoses in dermatology. In this application background skin color/tone (an unaffected patch of skin) is often known, as well as information about the affected area. Both can be used to help detect deviation. In Chapter 4 we explore the application of measuring reddening including the size of a wheal (raised patch) after allergy testing and detecting skin lesions including melanoma and other deadly forms of skin cancers.

In this chapter we present background on color images and discuss existing methods for processing these color images. In Section 2.1 we review color images and different representations of color images. In Section 2.2 we discuss work on color edge detection, a popular image processing task that is often dependent on color.

In Section 2.3 we introduce two new DSGS methods based on finding a projection of the data in color space that best enhances the areas of interest. In Section 2.4 we introduce the use of support vector machines as a way to define the DSGS space. In Section 2.5 we discuss a basic method for segmenting the DSGS images based on thresholding (used in evaluating the new methods). In Section 2.6 we compare the presented DSGS methods on a set of test problems. Finally in Section 2.7 we summarize our findings and present conclusions.

## 2.1 Background

Capturing digital color images has become as easy as pulling out your smart phone and pressing a button. The result is a high resolution image ready for new uses and applications. These digital images are generally captured in RGB color space, a three primary color space (Red, Green and Blue). Given the digital nature of these images, there are many color space transformations that may aid in uses of these images. A thorough background on color spaces as they relate to digital images can be found in the standard textbooks, e.g. [3]. In this section we give an overview of how color images are represented digitally and discuss some of the alternative representations used for artistic as well as for scientific purposes.

### 2.1.1 Digital Images

Today, most images are recorded digitally, capturing a continuous scene by storing a discrete numerical representation of the scene. The numerical represen-



tation is made up of a collection of pixel values, where resolution is determined by the number of pixels. Depending on the format of the image, the pixel values can be 1-bit (binary, for black and white image), 8-bit integers, 32-bit integers or floating point numbers. Most commonly, grayscale pixels are represented by 8-bit integers with values ranging from 0 (black) to 255 (white), with values in between representing shades of gray. To represent typical color images, each pixel has three coordinates which are stored in three separate color arrays. These are combined for display. When printed, colors are displayed as an additive combination of the coordinates. When projected, colors are displayed as a subtractive combination of the coordinates. Color images are often 24-bit (three 8-bit values for each pixel) resulting in  $2^{24}$  possible distinct colors.

### 2.1.2 Standard Representations for Color Images

There are a number of different formats for storing and displaying color images. Most users are familiar with color spaces such as **RGB** (**R**ed, **G**reen, **B**lue), which is generally used for displaying colors on computers, and **CMYK** (**C**yan, **M**agenta, **Y**ellow, **B**lack), which is generally used for printing. Each digital color seen on a screen or in print is a mixture of three primary colors. **K** in **CMKY** stands for black, which could be represented within **C**, **M** and **Y**, but is instead represented with **K** to save ink.

These color schemes were developed experimentally. For example, a human was presented with a fixed wavelength color to the left eye and was given **R**, **G**



Figure 2.2: A color image broken into its red, green, and blue coordinates, along with the grayscale version.

and **B** adjusting knobs to adjust to match that color for the right eye [3]. These experiments were first done in the late 1920s by William David Wright to define the 1931 **CIEXYZ** color space. **CIEXYZ** color space is a three primary color space that is the standard for converting between other color spaces. Although the three primary color space red, blue, green matches our understanding of the physics of color, it does not match how we perceive color. Characteristic features such as shade or even the lightness or darkness are correlated between each of the color channels. In Figure 2.2, one can see an image where each color plane or coordinate is displayed separately. From these separate images alone it would be challenging to discern different colors, or even the lightness or darkness, as these features are correlated among the color coordinates.

There are alternatives to the three primary color spaces. Alternative color spaces often try to match our perceived or intuitive understanding of how colors relate. The relationship between these different color spaces can be linear or non-linear and often are not one-to-one. The different color spaces are compared by their *gamut*. The *gamut* is defined as the subset of visible colors that can be displayed by a particular color space. The *gamut* of **RGB** is much smaller than the *gamut* of all

visible colors.

### 2.1.3 Representation of Grayscale Images

Most digital images are captured in color and then converted to a grayscale (**GS**) image. Converting a color image to a **GS** image is based on a model of shooting a photo in black and white in which the film measures the intensity of light. MATLAB's `rgb2gray.m` converts a color image to **GS** using the formula

$$X = 0.2989\mathbf{R} + 0.5870\mathbf{G} + 0.1140\mathbf{B}, \quad (2.1)$$

where each **GS** pixel is determined by a linear combination of the values of **R**, **G** and **B**. Unfortunately, the resulting grayscale images lose crucial information about color and how one color relates to another. Consider the image of M&Ms in Figure 2.3. One can quickly distinguish and count the red M&Ms in the color image, but in the **GS** image the red and blue M&Ms are visually indistinguishable. In short, the **GS** model does not capture our perception of the difference between red and blue.

### 2.1.4 Other Representation of Color Images

In addition to the above mentioned color spaces there have been a number of other color spaces suggested both for artistic and scientific purposes. In this section we review two classes of color spaces, one based on cylindrical coordinates and one based on how humans perceive color (opponent color spaces).



Figure 2.3: A color image of M&Ms (left) and the grayscale image (right) computed using MATLAB's `rgb2gray.m`. Source: <http://blogs.mathworks.com/images/steve/2010/mms.jpg>.

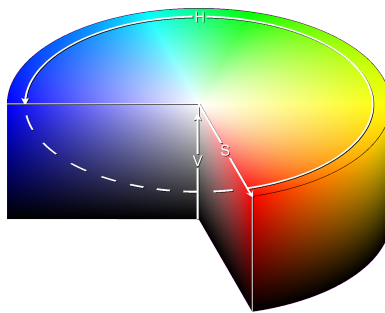


Figure 2.4: Schematic of the **HSV** color space. Source: Image created by (3ucky(3all using Borland Delphi 2006, Adobe Photoshop 9.0. Source: [http://upload.wikimedia.org/wikipedia/commons/e/e0/HSV\\_cylinder.png](http://upload.wikimedia.org/wikipedia/commons/e/e0/HSV_cylinder.png).

#### 2.1.4.1 Cylindrical Coordinate Color Spaces

There are a number of color spaces which use cylindrical coordinates. Many people consider this approach more intuitive than the three primary color spaces [3]. Generally, the angle measures the *Hue* (color), the distance from the central axis represents the *Saturation* of the color, and the distance along the axis measures the *Lightness*, also known as *Value* or *Brightness*. Color spaces in this class include **HSV**, **HSI** or **HSL**. A schematic of the **HSV** color space is given in Figure 2.4.

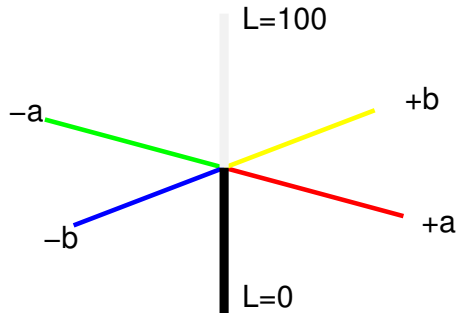


Figure 2.5: Schematic of the 1976 CIE Lab color space.

#### 2.1.4.2 Opponent Color Spaces

Opponent color spaces were developed to match the human perception of color. The goal of these color spaces is to find a plane that is uniform in *chroma* (color) and an axis that is uniform in *luminance* (light intensity). These spaces are based on the theory that humans perceive red and green, as well as blue and yellow, as opposites [3]. These color spaces are generally non-linearly related to the three primary color spaces.

The 1976 **CIE Lab** color space is a standard opponent-color space, used in a variety of photo editing software systems and in some image file types (including ‘.tiff’ and ‘.pdf’ image files). In **CIE Lab** the coordinates are:

- an  $L$  coordinate, measuring the *luminance*,
- a *chroma* plane, where the  $a$  axis scales between *green* (negative) and *red* (positive) and the  $b$  axis scales between *blue* (negative) and *yellow* (positive) as shown in Figure 2.5.

The transformation to **CIELab** from the 1931 **CIEXYZ** space is given by

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} 0 & 116 & 0 \\ 500 & -500 & 0 \\ 0 & 200 & -200 \end{bmatrix} \begin{bmatrix} f(X/X_n) \\ f(Y/Y_n) \\ f(Z/Z_n) \end{bmatrix} + \begin{bmatrix} -16 \\ 0 \\ 0 \end{bmatrix}, \quad (2.2)$$

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3, \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{otherwise,} \end{cases}$$

where  $X_n, Y_n, Z_n$  is the reference white point; there are many standards for the reference white point based on the device displaying the image [3]. Converting between **RGB** and **CIELab** is time consuming [19] and the gamut of **CIELab** is much larger than that of **RGB**.

### 2.1.5 Modified Color Spaces for Specific Applications

We briefly discuss two examples of work on developing modified color spaces to perform specific image processing tasks on color images.

#### 2.1.5.1 Skin Color Models

There is substantial research in user-aided methods for detecting skin in both color images and video. There have been several approaches to finding a model of the skin, based on a user-provided information and followed by thresholding the image based on the characteristics in the model. Yang et al. [92] first find a normalized 2D red-green space from the **RGB** color space using the transformation

$$r = \frac{R}{R + G + B}, g = \frac{G}{R + G + B}. \quad (2.3)$$

Then they compute the mean and the covariance matrix for pixels in a user-identified skin patch. Modeling the patch using a normal distribution, they classify other pixels as skin or non-skin.

#### 2.1.5.2 Pipe Image Interpretation Based on Support Vector Machines

In this application the goal is to detect defects in pipes. Mashford et al. [50] present a method of building a learning set based on **RGB** information that is supplied. In this case, they are segmenting between unaffected pipes and features such as joints, pipe connections and defects. They use support vector machines (details follow in Section 2.4) that can learn from two classes of information. This results in a binary color space that aids discriminating between the two classes.

### 2.2 Previous Work in Color Edge Detection

Many image processing tasks rely on well established image processing algorithms such as edge detection algorithms. There have been several types of color edge detection methods suggested. One class of algorithms, described in Section 2.2.1, decomposes images by processing each coordinate separately, similar to a grayscale image. Another class of algorithms, described in Section 2.2.2, is vector based, where the pixel is treated as a vector throughout the computation. In addition to the two main classes, research has been done comparing color histograms of patches to determine whether an edge exists between two patches [49, 74]. Median principal component analysis is used to find edges [22, 66].

All of these methods are significantly more costly than using a standard edge detection method on a grayscale image. Depending on what color space is used and what is being detected, the benefits over using the grayscale image can be minimal.

### 2.2.1 Decomposed Edge Detection

Decomposed edge detection was first suggested in 1977 [58] for **RGB** images and more recently has been suggested for other color spaces including **HSI** [9] and **CIELab** [29]. There are two suggested decomposed edge detection approaches. One computes the gradient magnitude for each coordinate and combines the gradient information into a single gradient mask which is then used to find the edges. The other finds edges in each coordinate and combines the edge information. Decomposed edge detection methods are easily implemented.

### 2.2.2 Vector Based Edge Detection

Vector based edge detection methods use the pixel vector information in order to find the edge map. Research in vector based methods include work by [53,82,83]. Vector based methods preserve the color by keeping the pixel vector information. The basic idea of these methods is to look at a window and determine if there exists an edge in that window. Below we describe vector based methods for an **RGB** image but it is easily adapted to any of the color spaces described.

Each pixel (a vector of size three) is compared to all other pixels in the window



by computing the Euclidean distance

$$d_m = |\mathbf{RGB}_{i,j} - \mathbf{RGB}_{k,l}| = \sqrt{(R_{i,j} - R_{k,l})^2 + (G_{i,j} - G_{k,l})^2 + (B_{i,j} - B_{k,l})^2}, \quad (2.4)$$

where

- $(i, j)$  is the index of the selected pixel and  $(k, l)$  is the index of another pixel in the window.
- for a  $3 \times 3$  window,  $m$  is an index from 1 to 9 for each of the 9 pixels.

Then we can compute a single value for each pixel as

$$d_{i,j} = d_1 + d_2 + d_3 + d_4 + d_5 + d_6 + d_7 + d_8 + d_9. \quad (2.5)$$

If the range of  $d_{i,j}$  values is larger than a user threshold, the pixel with the median value is said to be an edge. The basic method for computing the range is to compute the difference between the biggest  $d$ -value and the smallest  $d$ -value. There have been many different suggestions to compute the range in order to protect against noise including: Vector Range (VR) edge detection, Vector Dispersion (VD) edge detection, Mean Vector Range (MVR) edge detection and Mean Vector Dispersion (MVD) edge detection [83].

## 2.3 Projection to Find New Domain-Specific Grayscale Color Space

The inadequacies discussed above that arise when converting between standard color spaces, when considering the limitations of the grayscale images, and when including the costliness of color edge detection methods suggest a need for a new single coordinate space that is fast to compute and retains the color information of interest for the specific image processing task at hand.

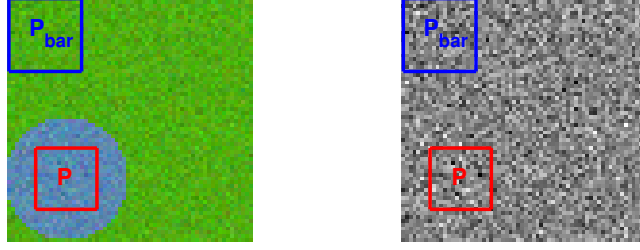


Figure 2.6: Left: Synthetic image created in MATLAB with two contrasting colors that appear to be the same color in the **GS** image. Right: **GS** image found using `rgb2gray.m`. The blue and red boxes mark the user selected background ( $\bar{P}$ ) and foreground ( $P$ ).

Consider Figure 2.6, a noisy color image with a uniform background color and a large dot of a contrasting color (left) with its corresponding **GS** image (right). Consider the task of detecting the edges around the blue dot. In the **GS** image this task would be impossible, because the colors map to the same gray level. Now, consider the scatter in Figure 2.7. Each pixel contained in the blue box (background) is represented by blue dots, and each pixel contained in the red box (foreground) is represented by red dots. In the scatter plot, one can easily see that there is an orientation that would best distinguish the difference between the colors. If we rotate our data we can find a rotated coordinate that best distinguishes between the background and foreground, as shown in Figure 2.8. Using the coordinates along the  $\mathbf{R}'$  axis, we obtain the image in Figure 2.9.

This motivates the development of a new color space, analogous to the grayscale (single coordinate), adapted to the image and the task. We do this by finding a transformation of the **RGB** image into a single coordinate that highlights the areas of interest and makes the task as easy as possible. Using the user-defined

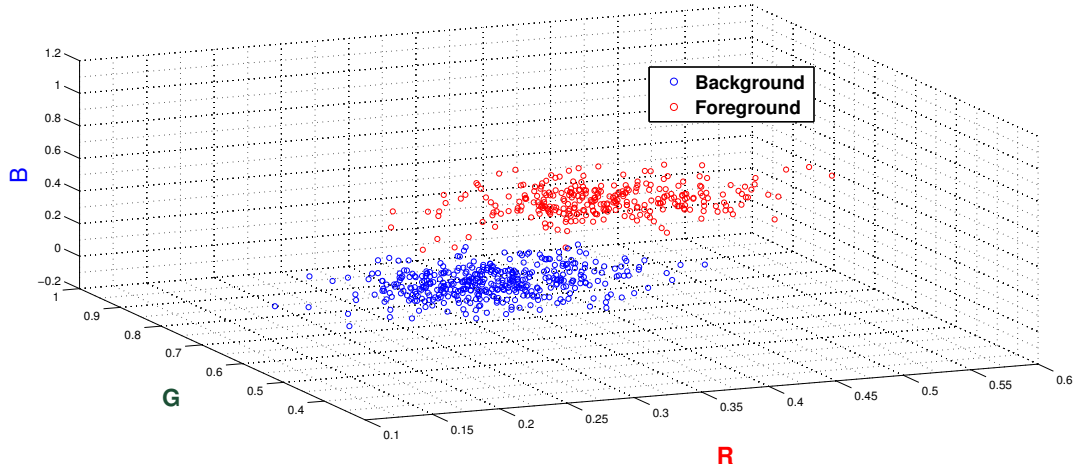


Figure 2.7: Scatter plot of the pixels in the background ( $\bar{P}$ , blue) and the pixels in the foreground ( $P$ , red).

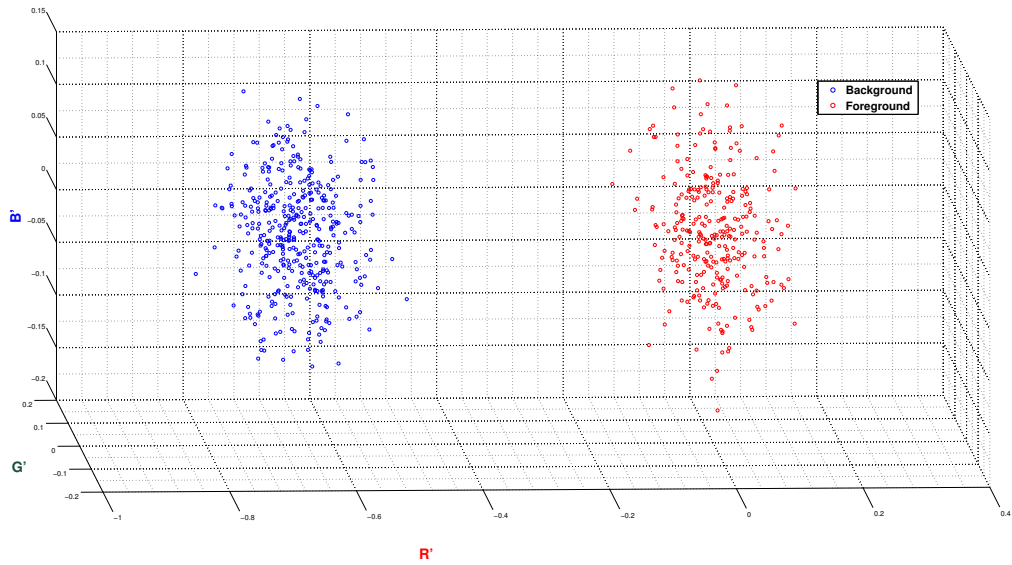


Figure 2.8: Scatter plot of the pixels in the background ( $\bar{P}$ , blue) and the pixels in the foreground ( $P$ , red) after rotation.

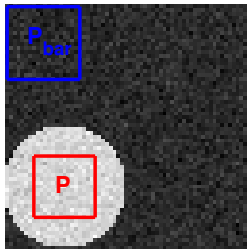


Figure 2.9: Resulting image after rotating the image and projecting onto the coordinate  $\tilde{\mathbf{R}}$  from Figure 2.6.

background and foreground (object of interest) we find the transformed space that best highlights the differences between the background and foreground. We present methods for finding linear (this chapter) and non-linear (next chapter) transformations of the color image to project onto a single coordinate. We call this new class of transformation methods the Domain-Specific Grayscale (DSGS) methods.

For the linear DSGS we present a method very similar to that described in the motivating example: we simply find the orientation that produces the greatest differences between each of the pixels in foreground and background, or, alternatively, the orientation that minimizes the overlap between the distributions of the background and foreground patches.

### 2.3.1 Computing the Projection

We represent a standard color image as  $\mathbf{X}$  where  $\mathbf{X} \in \mathbb{R}^{m \times n \times 3}$  for an  $m \times n$  pixel image. Alternatively, we represent the image by  $\mathbf{x} \in \mathbb{R}^{mn \times 3}$  and pixel  $i$  is represented by  $\mathbf{x}_i = [x_i^1, x_i^2, x_i^3]$ .

We assume that the user has designated patches that define the foreground

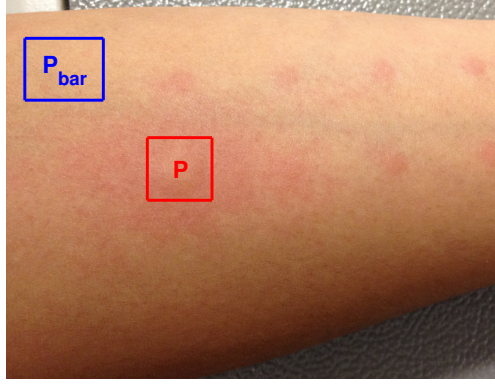


Figure 2.10: Image of a patient's skin after an allergy test is administered. Source: Rho Inc.

and background of interest . We call the foreground patch  $\mathbf{P}$  and background patch  $\bar{\mathbf{P}}$ , where  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  are non-overlapping submatrices of  $\mathbf{X}$ .

As an example, consider the image of skin after allergy test as seen in Figure 2.10. A physician could easily identify a patch of unaffected skin ( $\bar{\mathbf{P}}$ ) and a patch of affected skin ( $\mathbf{P}$ ). We discuss this example in more detail in Chapter 4.

Our goal is to find a  $\mathbf{w} \in \mathbb{R}^3$  such that the pixels in our new single coordinate image can be represented by  $y_i = \langle \mathbf{x}_i, \mathbf{w} \rangle$  . For ease of visually understanding this transformation we consider rotating the space so that the rotated first coordinate maximizes the distance. Rotation around the  $\mathbf{RG}$  plane is given by

$$\mathbf{T}_{\mathbf{RG}}(\alpha) = \begin{bmatrix} \cos(\theta_{\mathbf{RG}}) & 0 & -\sin(\theta_{\mathbf{RG}}) \\ 0 & 1 & 0 \\ \sin(\theta_{\mathbf{RG}}) & 0 & \cos(\theta_{\mathbf{RG}}) \end{bmatrix}, \quad (2.6)$$

and rotation around the  $\mathbf{RB}$  plane is given by

$$\mathbf{T}_{\mathbf{RB}}(\beta) = \begin{bmatrix} \cos(\theta_{\mathbf{RB}}) & -\sin(\theta_{\mathbf{RB}}) & 0 \\ \sin(\theta_{\mathbf{RB}}) & \cos(\theta_{\mathbf{RB}}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

where  $\theta_{\mathbf{RG}}$  is the angle that the  $\mathbf{R}$  axis is rotated in the  $\mathbf{RG}$  plane and  $\theta_{\mathbf{RB}}$  is the angle that the  $\mathbf{R}$  axis is rotated in the  $\mathbf{RB}$  plane. We are then interested in the orthogonal transformation

$$\mathbf{T} = \mathbf{T}_{\mathbf{RG}} \mathbf{T}_{\mathbf{RB}}, \quad (2.8)$$

and  $\mathbf{w} = \mathbf{T}(1, :)$ .

### 2.3.1.1 Maximizing Distance between Pixels (MDP)

In this case we are interested in finding the direction that maximizes the distance between the patches, so we solve the problem

$$\max_{\|\mathbf{w}\|=1} \left( \sum_{i \in \mathbf{P}} \sum_{j \in \bar{\mathbf{P}}} y_j - y_i \right)^2 \quad (2.9)$$

where  $y_i = \langle \mathbf{x}_i, \mathbf{w} \rangle$ . We can write the new image  $\mathbf{y} \in \mathbb{R}^{mn}$ .

We call this DSGS method Maximize Distance between Pixels (MDP). MDP is intuitive in nature but not always effective. In some applications the solution to (2.9) may not cluster the foreground and background pixels together. Consider the example where the foreground pixels are clustered around zero and the background is split between  $[-1, -1, -1]$  and  $[1, 1, 1]$ .

### 2.3.1.2 Minimizing Area between Distributions (MAD)

The MDP method maximizes the sum of squared distances between  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  but the variation within the samples might be much larger than the difference between the samples. In MAD we try to balance the distance between the pixels

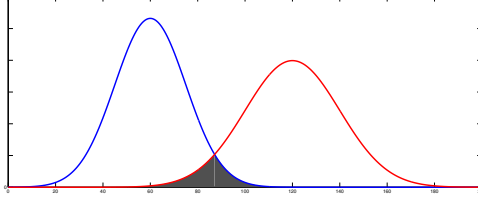


Figure 2.11: The area in gray is the area over which we are minimizing.

and the standard deviation within each patch.

We use a simple model of each patch. We assume that the original pixel values in  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  remain normal distributions. Then after projection by  $\mathbf{w}$ , since normal distributions are rotationally invariant [5], the pixels are also normally distributed with means  $\mu_w^{\mathbf{P}}$  and  $\mu_w^{\bar{\mathbf{P}}}$  and standard deviations  $\sigma_w^{\mathbf{P}}$  and  $\sigma_w^{\bar{\mathbf{P}}}$ . The density function for a normal distribution with mean  $\mu$  and standard deviation  $\sigma$  is given by

$$f(y, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right). \quad (2.10)$$

Figure 2.11 shows the typical case, after projection. We would like a projection  $\mathbf{w}$  for which the shaded region in Figure 2.11 is minimized. We solve for the intersection point  $z$  of the densities, located between  $[\mu_w^{\bar{\mathbf{P}}}, \mu_w^{\mathbf{P}}]$ , by solving

$$f(z, \mu_w^{\bar{\mathbf{P}}}, \sigma_w^{\bar{\mathbf{P}}}) = f(z, \mu_w^{\mathbf{P}}, \sigma_w^{\mathbf{P}}), \text{ for } z \in [\mu_w^{\bar{\mathbf{P}}}, \mu_w^{\mathbf{P}}]. \quad (2.11)$$

In the case where there is no orientation where  $z \in [\mu_w^{\bar{\mathbf{P}}}, \mu_w^{\mathbf{P}}]$ , we set the intersection to be the midpoint between  $\mu_w^{\bar{\mathbf{P}}}$  and  $\mu_w^{\mathbf{P}}$ . If this is the case, the intersecting area will be large ( $> 0.5$ ) and non-optimal and the resulting image will poorly differentiate between the background and foreground.

The shaded region has area

$$1 + (\text{sign}(\mu_w^{\bar{\mathbf{P}}} - \mu_w^{\mathbf{P}})) \int_{-\infty}^z f(y, \mu_w^{\bar{\mathbf{P}}}, \sigma_w^{\bar{\mathbf{P}}}) dy + (\text{sign}(\mu_w^{\mathbf{P}} - \mu_w^{\bar{\mathbf{P}}})) \int_{-\infty}^z f(y, \mu_w^{\mathbf{P}}, \sigma_w^{\mathbf{P}}) dy \quad (2.12)$$

where

$$\int_{-\infty}^{\infty} f(y, \mu, \sigma) dy = 1. \quad (2.13)$$

We minimize (2.12) over all choices  $\mathbf{w}$  where  $\|\mathbf{w}\| = 1$ . We can write the new image as  $\mathbf{y} \in \mathbb{R}^{mn}$ .

We call this DSGS method Minimizing Area between Distributions (MAD).

## 2.4 Use of Support Vector Machines to Determine a New Color Space

A support vector machine (SVM) is a machine learning algorithm partitioning a set into two subsets. It was developed by Vladimir N. Vapnik as a “non-probabilistic linear classifier” [77]. As a machine learning algorithm it takes in a training set with known classifications and learns from that set in order to classify the entire set. In image processing tasks, SVM has been used extensively for image classification [77] as well as for color image segmentation [33, 50].

SVM classifies the data by using a score based on the distance from the hyperplane. We suggest using the score as the new pixel value. The resulting single-coordinate image results from an orientation that separates the training sample. In this section we introduce a new view of SVM as a DSGS method.

### 2.4.1 Hard Margin SVM

SVM determines a hyperplane that separates the two classes (background/foreground) and maximizes the distance from the separating hyperplane to the data. This is the *maximum-margin hyperplane*.



In our example the training set will be the user supplied  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  and the classification “foreground” or “background.” Define

$$\mathcal{D} = \{(\mathbf{p}_i, u_i) | \mathbf{p}_i \in \bar{\mathbf{P}} \cup \mathbf{P}, u_i = 1 \text{ if } \mathbf{p}_i \in \mathbf{P} \text{ and } u_i = -1 \text{ if } \mathbf{p}_i \in \bar{\mathbf{P}}\}_{i=1}^k, \quad (2.14)$$

where  $k$  is the number of pixels in  $\bar{\mathbf{P}} \cup \mathbf{P}$ . We represent our hyperplane as

$$\langle \mathbf{p}, \mathbf{w} \rangle - b = 0. \quad (2.15)$$

If the elements of  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  are separable from each other then we can find an infinite number of hyperplanes such that

$$\langle \mathbf{p}_i, \mathbf{w} \rangle - b \geq u_i, \text{ if } u_i = +1 \quad (2.16)$$

and

$$\langle \mathbf{p}_i, \mathbf{w} \rangle - b \leq u_i, \text{ if } u_i = -1, \quad (2.17)$$

which we can write as

$$u_i(\langle \mathbf{p}_i, \mathbf{w} \rangle - b) \geq 1. \quad (2.18)$$

Then our problem is to find the maximum-margin hyperplane by finding  $\mathbf{w}$  and  $b$  that maximizes the distance between the hyperplane and each of the two classes. The distance between the pixels on the margins and the hyperplane that separates the two classifications is  $\frac{1}{\|\mathbf{w}\|}$ , and we want to maximize this. The schematic of the geometry of the hyperplane in  $2D$  is given in Figure 2.12. We can state the problem as an optimization problem

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2^2$$

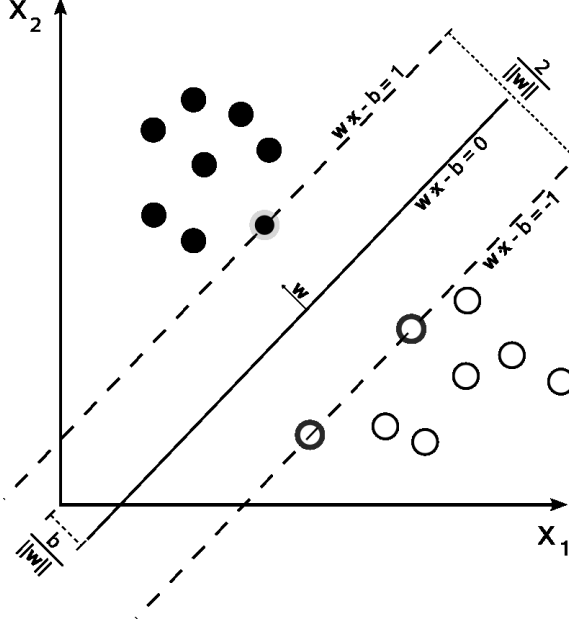


Figure 2.12: Schematic of hard margin SVM where the two classes of data are represented by the filled and empty dots respectively. Source: [http://upload.wikimedia.org/wikipedia/commons/2/2a/Svm\\_max\\_sep\\_hyperplane\\_with\\_margin.png](http://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png).

$$\text{s.t. } u_i(\langle \mathbf{p}_i, \mathbf{w} \rangle - b) \geq 1 \text{ for } i = 1, \dots, k. \quad (2.19)$$

This problem can be solved using standard quadratic programming techniques [35].

Once we find  $\mathbf{w}$ , the score or distance from the hyperplane is given by

$$y_i = \langle \mathbf{x}_i, \mathbf{w} \rangle - b \quad (2.20)$$

### 2.4.2 Soft Margin SVM

SVM provides a powerful tool for separating data. Unfortunately in a real world example the data may not be separable. In this case, the hard margin SVM problem (2.23) will fail to have a solution. This can easily happen if  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  have significant noise or misclassification errors. Noise is inherent in images. Errors can be introduced because of artifacts or blurring in the image. A user could introduce

errors when selecting  $\mathbf{P}$  and  $\bar{\mathbf{P}}$ , for example, by accidentally including pixels that are not part of the area of interest. Therefore the *hard margin* SVM may fail either because the samples are not linearly separated or because errors or noise skew the resulting hyperplane. In both of these cases it is evident that the *hard margin* condition needs to be relaxed to allow some of the erroneous data points to be “misclassified” leading to a *soft-margin* SVM.

Soft-margin SVM introduces a slack variable  $\xi_i \geq 0$  for each data point. The problem then becomes balancing the margin and the slack variable. We use  $C > 0$  as a penalty parameter, controlling the trade-off between these two objectives. A *soft-margin* SVM is determined by

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \|\boldsymbol{\xi}\|_1 \quad (2.21)$$

$$\text{s.t. } u_i(\langle \mathbf{p}_i, \mathbf{w} \rangle - b) \geq 1 - \xi_i \quad (2.22)$$

$$\xi_i \geq 0 \text{ for } i = 1 \dots k. \quad (2.23)$$

As  $C$  increases, (2.23) moves more towards a *hard-margin* SVM. A  $C$  closer to zero allows for more misclassifications.

## 2.5 Thresholding Images for Segmentation

To evaluate MDP, MAD, and SVM Domain Specific Grayscale methods, we will use the more primitive task of classifying the pixels of the image as background (of value zero) or as foreground (of value one), resulting in a binary image. Basic classification or segmentation methods are based on thresholding the image based on some threshold level  $T$ .

For SVM the threshold value is simply  $T_{SVM} = 0$ , since the SVM score equals zero on the separating hyperplane. Similar to SVM, the MAD threshold can be set to the intersection point between distributions, where  $T_{MAD} = (\text{sign}(\mu_w^P - \mu_w^{\bar{P}}))z$ . Values  $y_i$  less than  $T_{MAD}$  would be assigned to the background and values greater than or equal to  $T_{MAD}$  would be assigned to the foreground.

For the other methods, including for transformation to grayscale, there is no natural threshold. A simple method would be to set the threshold as the midpoint between the means of  $\mathbf{P}_w$  and  $\bar{\mathbf{P}}_w$ :

$$T = \frac{1}{2} \text{sign}(\mu_w^P - \mu_w^{\bar{P}})(\mu_w^P + \mu_w^{\bar{P}}). \quad (2.24)$$

Unfortunately, if the standard deviations of  $\mathbf{P}_w$  and  $\bar{\mathbf{P}}_w$  are significantly different in size, (2.24) is likely to be a poor threshold. To account for this, we set the threshold to the point of equal probability of belonging to  $\mathbf{P}_w$  and  $\bar{\mathbf{P}}_w$ , assuming they are normally distributed.

The probability that a random variable  $Z$  is less than or equal to  $z$  for a normal distribution of mean  $\mu$  and standard deviation  $\sigma$  is given by

$$P(Z \leq z) = \int_{-\infty}^z f(y, \mu, \sigma) dy, \quad (2.25)$$

where  $f(y, \mu, \sigma)$  is the probability density function (2.11). Without loss of generality assume  $\mu_w^{\bar{P}} > \mu_w^P$ . Then we can find  $z$  such that

$$P(Z_{\bar{\mathbf{P}}} \geq z) = P(Z_{\mathbf{P}} \leq z) \quad (2.26)$$

where  $Z_{\bar{\mathbf{P}}}$  and  $Z_{\mathbf{P}}$  are the random variable the associated distributions for  $\bar{\mathbf{P}}_w$  and

$\mathbf{P}_w$  respectively. Or equivalently we can write

$$1 - \int_{-\infty}^z f(y, \mu_w^{\bar{\mathbf{P}}}, \sigma_w^{\bar{\mathbf{P}}}) dy = \int_{-\infty}^z f(y, \mu_w^{\mathbf{P}}, \sigma_w^{\mathbf{P}}) dy. \quad (2.27)$$

In general the threshold is given by  $T = (\text{sign}(\mu_w^{\mathbf{P}} - \mu_w^{\bar{\mathbf{P}}}))z$ .

## 2.6 Comparing the Linear Domain-Specific Grayscale Methods

In this section we present results comparing the linear DSGS methods. Tests are done on both synthetic images and natural images. For this work we only consider the case of segmenting between background and foreground. Methods are evaluated based on the percent of pixels misclassified compared to a ground truth segmentation (binary mask).

### 2.6.1 Natural Images

Our natural images come from three sources:

- MATLAB's images '*peppers.png*', '*football.jpg*' and '*saturn.png*'.
- A subset of images from the Berkeley Segmentation Dataset (Figure 2.13).
- Image of the skin after allergy testing courtesy of Rho Inc. (Figure 2.10).

#### 2.6.1.1 Berkeley Segmentation Dataset

The Berkeley Segmentation Dataset and Benchmark Dataset [48] is a collection of color images (and gray scale images) along with hand-drawn segments or

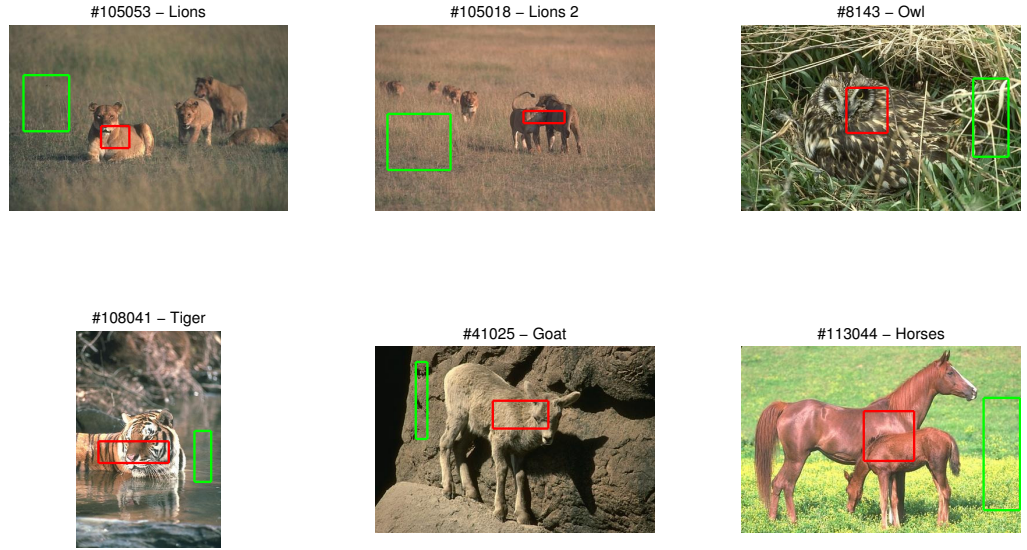


Figure 2.13: Subset of images from the Berkeley Segmentation Database.

edges.<sup>1</sup> We use a small subset of these images and simply classify background and foreground. A segmentation mask was created from the hand-drawn segment and assumed to be ground truth.

### 2.6.1.2 Synthetic Data

Synthetic images were created in two ways. The images titled ‘*Dot*’ and ‘*Sphere*’ were created by hand-selecting colors in the foreground and background. The remaining test images were created by taking samples from the natural image. A sample of the background and foreground was found by selecting patches from a natural image. A dot of the foreground was superimposed onto the background

---

<sup>1</sup>The use of the database for non-commercial research and educational purposes is free and encouraged “in the spirit of cooperative scientific progress” [48].

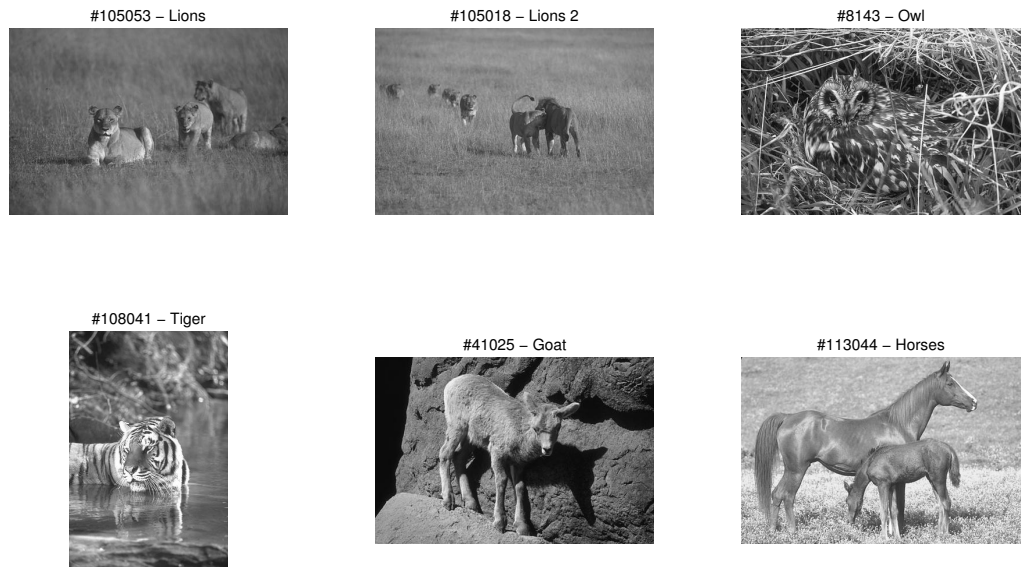


Figure 2.14: MATLAB's grayscale images for the Berkeley Segmentation Database images.

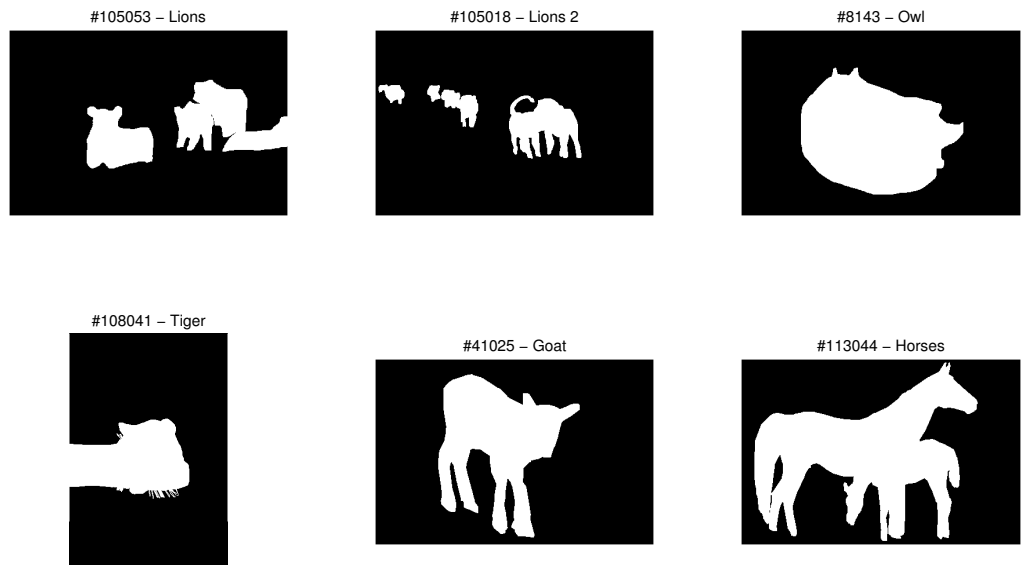


Figure 2.15: Human-user-filled edges for Berkeley Segmentation Database images.

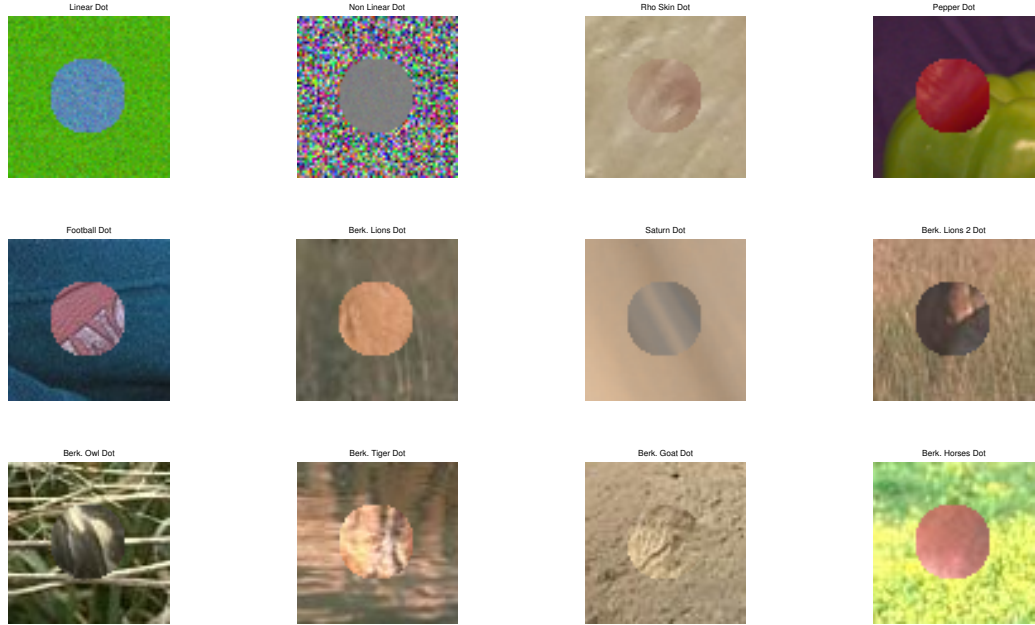


Figure 2.16: Synthetic data created from real images.

with a known segmentation mask. As seen in Figure 2.16, a circular patch of foreground is superimposed onto a patch of background. All these images are made to be  $64 \times 64 \times 3$  for ease of testing.

### 2.6.1.3 Implementation of Linear DSGS Methods

SVM was implemented using MATLAB's `svmtrain.m` using default parameters. The MAD and MDP optimization problems were solved using MATLAB's `fmincon` using the interior-point algorithm. All computations were done on a Mac OS X version 10.7.5 with 2.4 GHz Intel Core i7 processor.



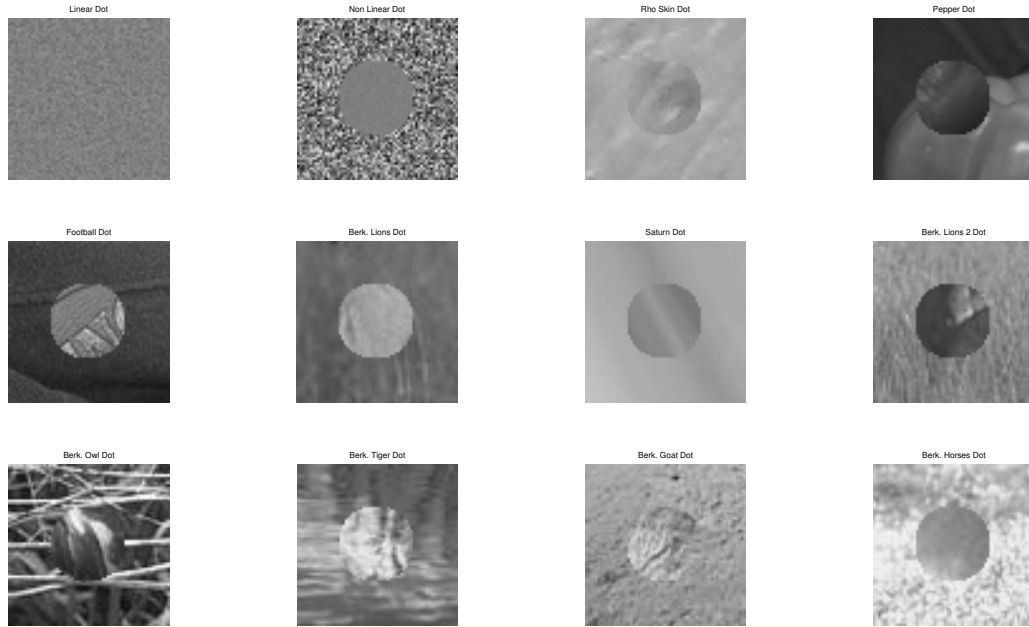


Figure 2.17: MATLAB's grayscale images for synthetic data created from real images.

## 2.6.2 Results on Synthetic Data

Figures 2.18 - 2.19 show the results after applying each of the methods on two different images, as well as the histogram of the results. As expected, all the methods do very well and produce very similar results on the synthetic image '*Dot*' as there is a clear orientation that best separates the two samples, as seen in the histogram of the pixels (Figure 2.7). In the example in Figure 2.19 of the image '*Lions 2 Dot*' the foreground and background pixels are not easily separated. In all these methods there is a varying degree of overlap between the foreground and background, as seen in the histogram of the results. In the example of '*Pepper Dot*' in Figure 2.20 the background is bimodal (made up of two distinct colors). All

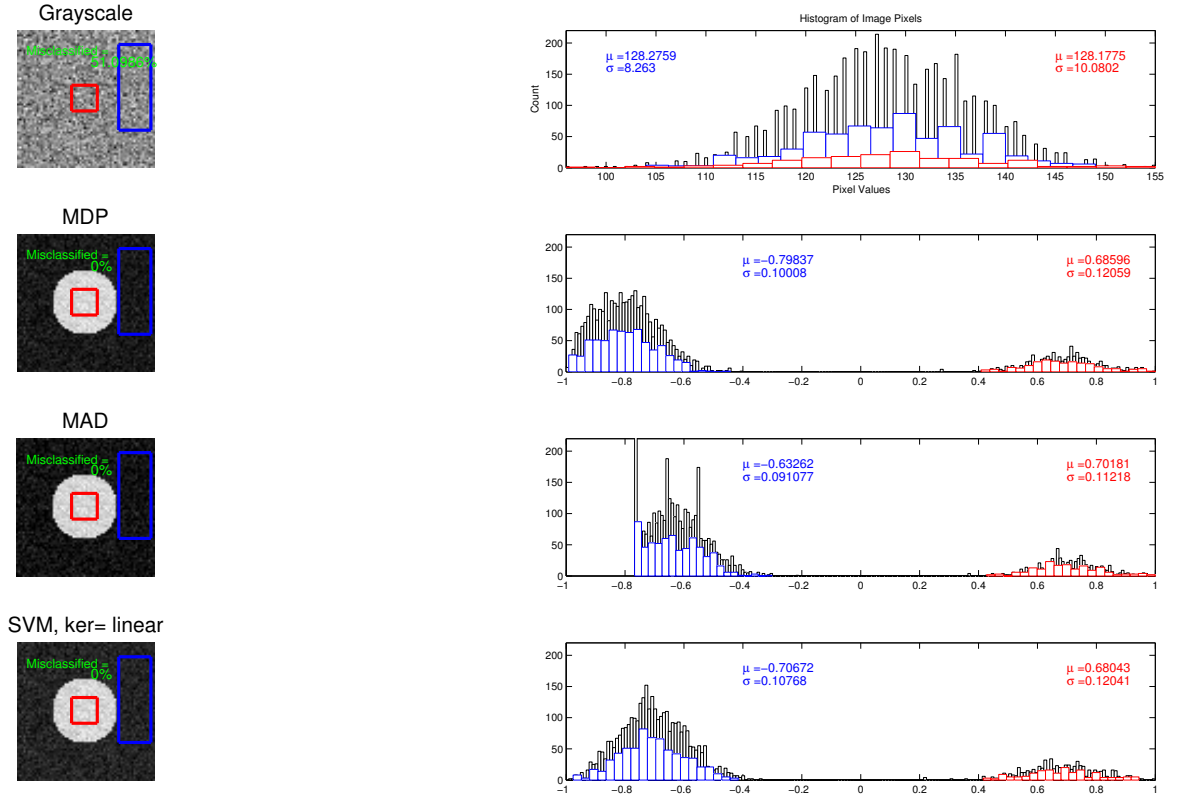


Figure 2.18: Resulting images after applying the different linear DSGS methods on the 'Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected  $P_w$  and  $\bar{P}_w$  in red and blue respectively.

three of the algorithms are able to separate the data but the resulting image has varying distributions of the background pixels. Variation within class (background or foreground) is important to consider as processing tasks such as edge detecting are more sensitive to these variations.

Table 2.6.2 shows the percentage of pixels misclassified for the twelve different synthetic images. The MAD and SVM methods clearly outperform the MDP method and grayscale. In a few examples the MDP does worse than grayscale for this task.

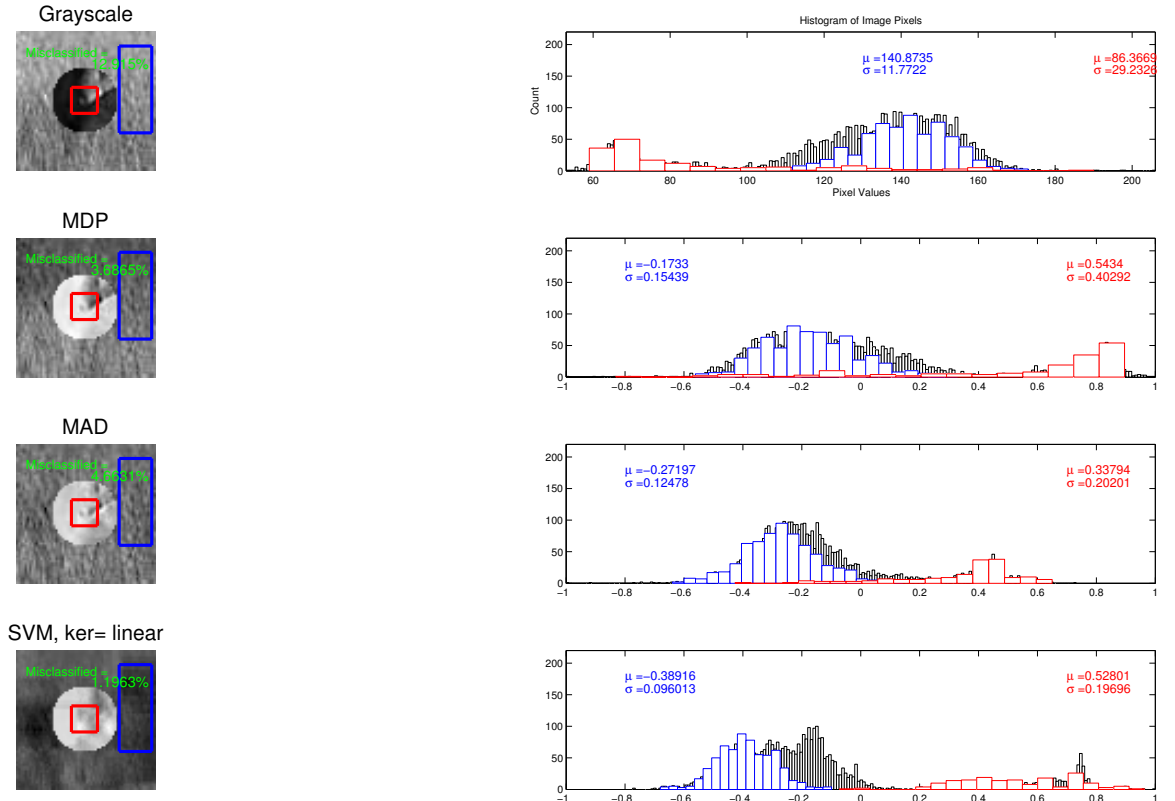


Figure 2.19: Resulting images after applying the different linear DSGS methods on the 'Lions 2 Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected  $P_w$  and  $\bar{P}_w$  in red and blue respectively.

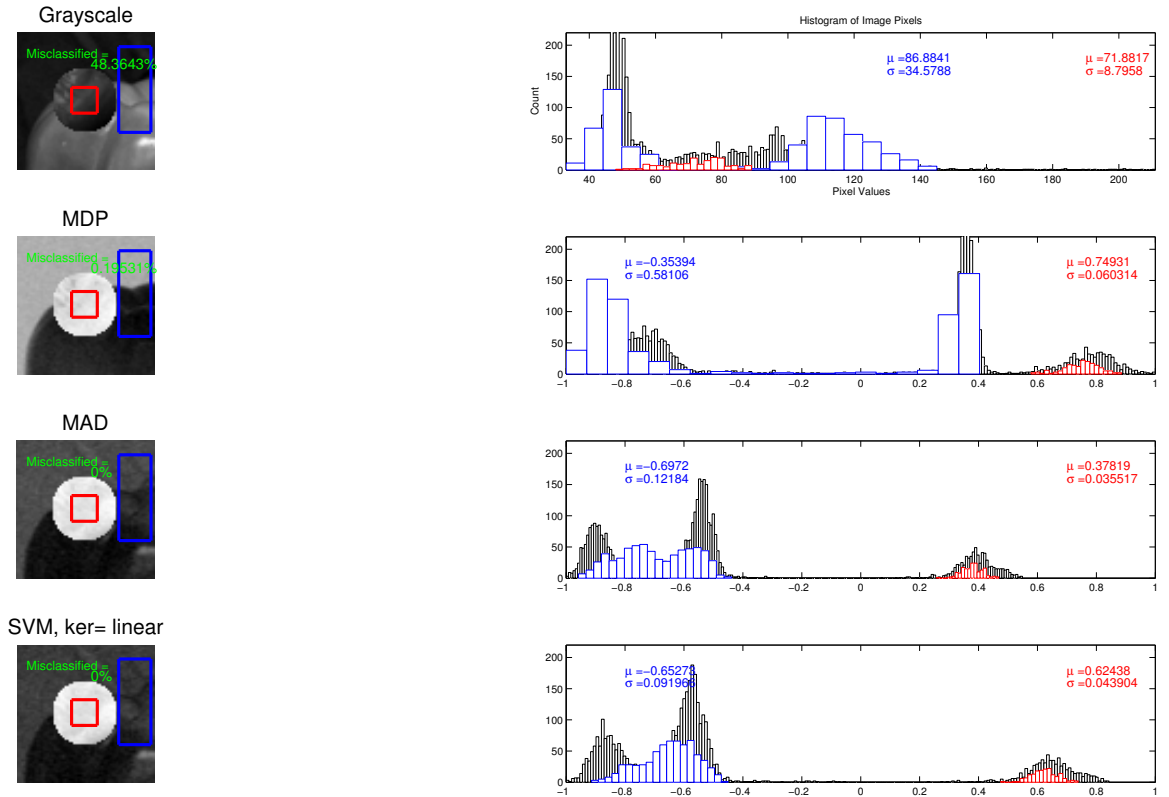


Figure 2.20: Resulting images after applying the different linear DSGS methods on the 'Pepper Dot' image along with the histogram of all the pixel values, with the pixels of the user-selected  $P_w$  and  $\bar{P}_w$  in red and blue respectively.

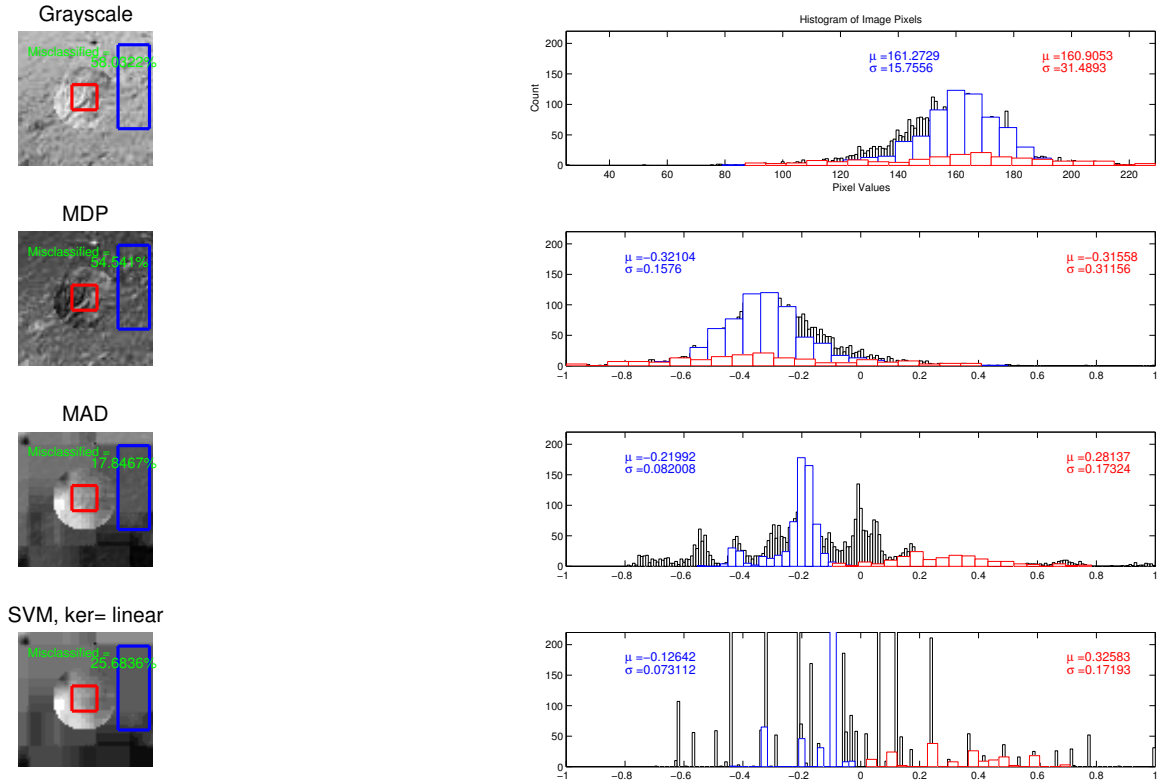


Figure 2.21: Resulting images after applying the different linear DSGS methods on the 'Goat Dot' image along with the histogram over all the pixel values with the pixels of the user selected  $P_w$  and  $\bar{P}_w$  in red and blue respectively.

Table 2.1: Percent misclassified for the synthetic images.

Image	GS	MDP	MAD	SVM, $C = 1$
Linear Dot	51.10	0	0	0
Non linear Dot	45.09	45.48	44.65	64.14
Skin Dot	7.28	7.20	0	0
Pepper Dot	48.36	11.77	0	0
Football Dot	1.98	0.63	0	0
Lions Dot	0.61	0.24	0.05	0
Saturn Dot	10.33	7.30	0.02	0.02
Lions 2 Dot	7.40	7.74	2.88	1.20
Owl Dot	41.28	41.38	16.53	12.60
Tiger Dot	16.28	15.92	15.92	15.41
Goat Dot	58.03	42.21	17.85	29.37
Horse Dot	15.31	4.88	0	0

### 2.6.3 Results on Natural Images

The human-drawn edges that are included with every image in the Berkeley data set were used to find a segmentation mask assumed to be ground truth. From Table 2.2 we can see that both MAD and SVM methods produce very similar results. For the images of the lions and the image of the horses, both methods produce excellent results. A visualization of the results of each method is given in Figures 2.22-2.23. For the other images, both methods improved upon the **GS** image but as seen in Figures 2.23 the results are not images that would be easily segmented by any algorithm. The image of the owl and the tiger have significant variation both in the background and foreground. In these cases nonlinear DSGS methods are needed.

Table 2.2: Percent misclassified for each image for a subset of Berkeley Segmentation Database.

Image	GS	MAD	SVM, $C = 1$
#105053 Lions	39.14	6.25	6.76
#105018 Lions 2	7.02	3.27	3.46
#8143 Owl	46.95	25.43	21.81
#108041 Tiger	41.29	24.55	18.50
#41025 Goat	33.74	31.73	27.55
#113044 Horse	18.75	2.20	2.50

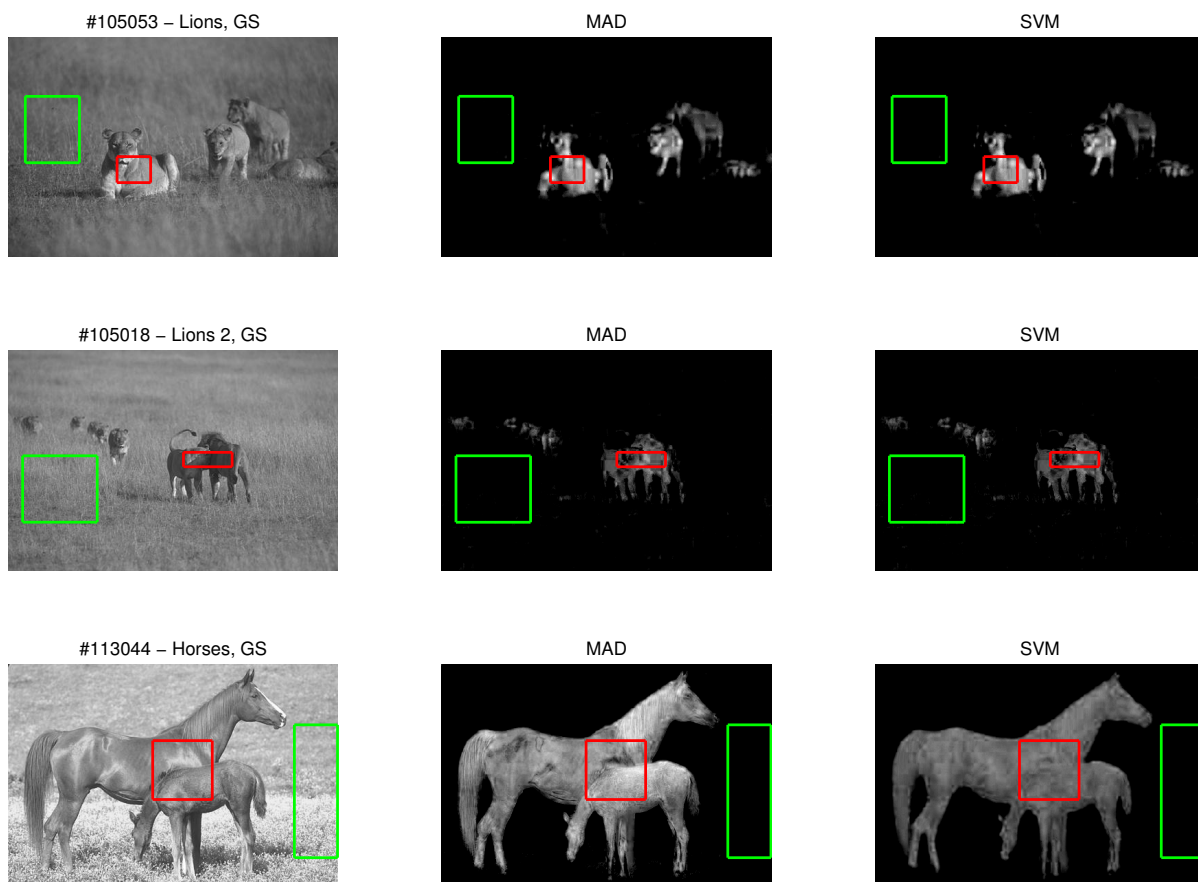


Figure 2.22: Comparing the **GS** image to the results of MAD and SVM for the images of lions and image of horses.

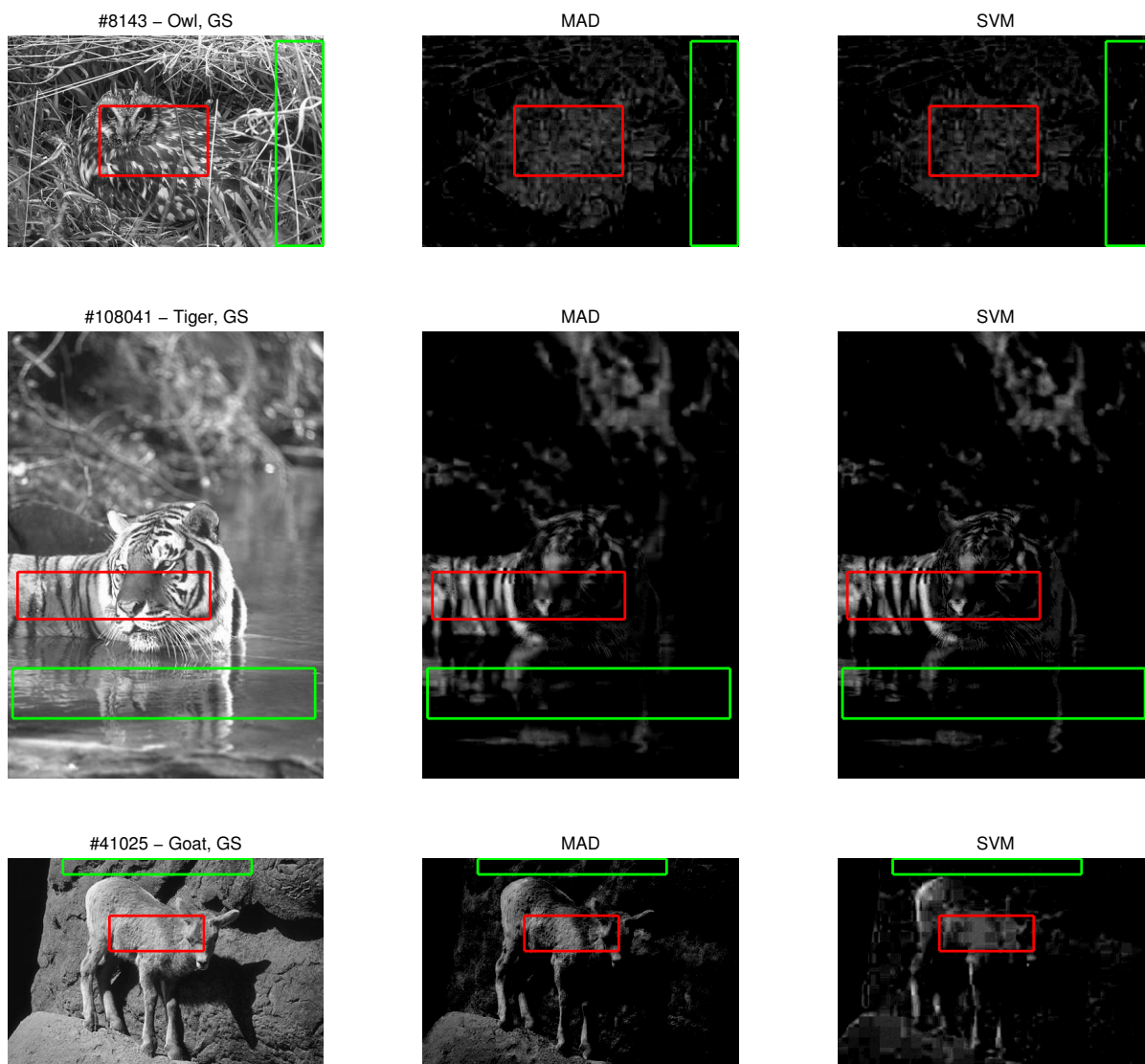


Figure 2.23: Image comparing the **GS** image to the results of MAD and SVM for the image of an owl, a tiger and a goat.



## 2.7 Summary and Conclusions

In this chapter we introduced a new class methods for finding a single-coordinate image from a color image based on user-provided domain information. We call this class of methods domain-specific grayscale (DSGS) methods. We developed the new MDP and MAD methods. The MDP method is based on finding a transformation that maximizes the distance between the pixels in the background and foreground, and the MAD method finds a transformation that minimizes the overlap between the distribution of the background and foreground pixels. In addition we introduced a new use of SVM, traditionally used as a classifier, to find a single coordinate image. We use the distance from the pixel to the SVM separating hyperplane as the pixel value for the DSGS image. In testing on the synthetic images and images from the Berkeley segmentation dataset, the MAD and SVM methods produced significantly better segmentations on the test images over the grayscale image. We found that the SVM and MAD methods produced very consistent and similar results. We choose to use the MAD method as a linear benchmark method because it is not reliant on a parameter such as  $C$  in the SVM method.

In Chapter 3 we will present nonlinear DSGS methods where pixels in background and foregrounds are not linearly related (for example the ‘*nonlinear dot*’).

## Chapter 3: Nonlinear Domain-Specific Grayscale Transformations for Color Images

If an image is made up of two distinct colors, then there exists a linear hyperplane that separates the two colors in **RGB** space, and the methods in the previous chapter work well. However, if the foreground and background are each comprised of many different colors, it becomes less likely that they can be linearly separated. Consider the image of a tiger in Figure 3.1. We compare the pixels of the selected background (blue box) and foreground (red box) in **RGB** space in Figure 3.2 and we can see that there exists no hyperplane that separates these regions. In this chapter we study methods for transforming a color image to a DSGS image when the separation between the foreground and background is not assumed to be linear.

In Section 3.1 we discuss the details of adapting SVM to address nonlinearly related data by mapping the data into a higher order space (which is implicitly done by applying a kernel), where the data is linearly related and then applying linear SVM. In Section 3.2 we show details on how to construct kernels to model the higher order space. In Section 3.3 we describe schemes for the selection of kernels and parameters. In Section 3.4 we propose the use of Kernel SVM as a nonlinear

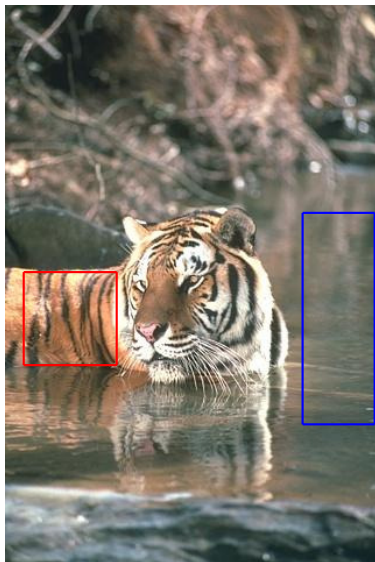


Figure 3.1: Image of a Tiger from the Berkeley Segmentation dataset with user selected background and foreground for training.

Domain-Specific Grayscale (DSGS) Method. A nonlinear DSGS method based on a new approach to learning the nonlinear relationship between the background and foreground is introduced in Section 3.5. In Section 3.6 we compare nonlinear DSGS methods on the test images from Chapter 2. In Section 3.7 we demonstrate some of the advantages of nonlinear methods even in cases where linear methods outperform the nonlinear methods in segmenting the test images.

### 3.1 Kernel SVM

Kernel SVM methods consist of two parts: mapping the data to a higher dimensional space where the data is more linearly related and applying a linear method to the mapped data.

We call the map to the higher order space the *feature map* which we define

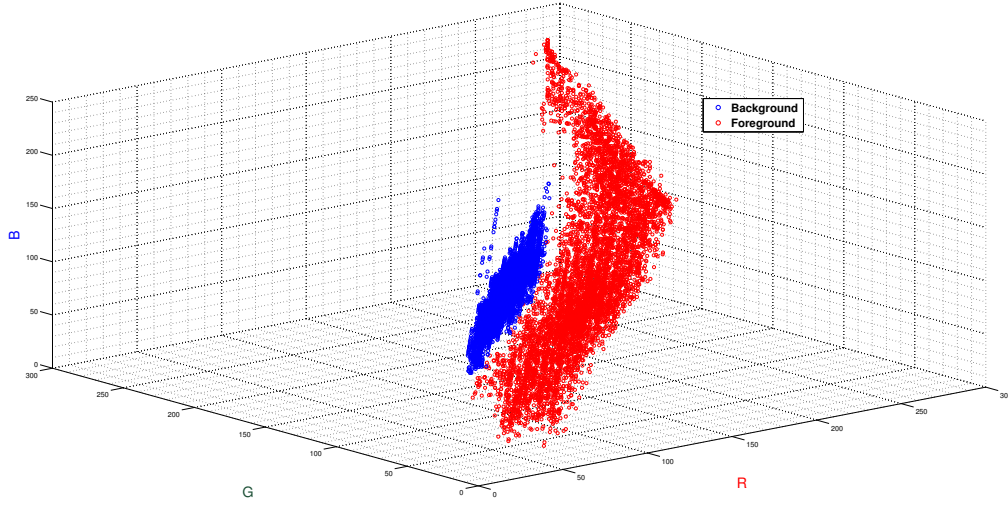


Figure 3.2: Selected background (blue box) and foreground (red box) pixels from Figure 3.1 plotted in **RGB** space.

as:

$$\Phi : \mathcal{X} \rightarrow \mathcal{H} \quad (3.1)$$

$$x \mapsto \Phi(x)$$

where  $\mathcal{X}$  is the pixel space and  $\mathcal{H}$  is the *feature space* [76, 77].

The usefulness of kernel methods is demonstrated by the following toy problem. Consider the two dimensional data pictured on the left in Figure 3.3. Clearly there is no separating hyperplane between the red and blue dots. If we transform our data into three dimensions using the map

$$\begin{aligned} \Phi : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ [x_1, x_2]^T &\mapsto [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T, \end{aligned} \quad (3.2)$$

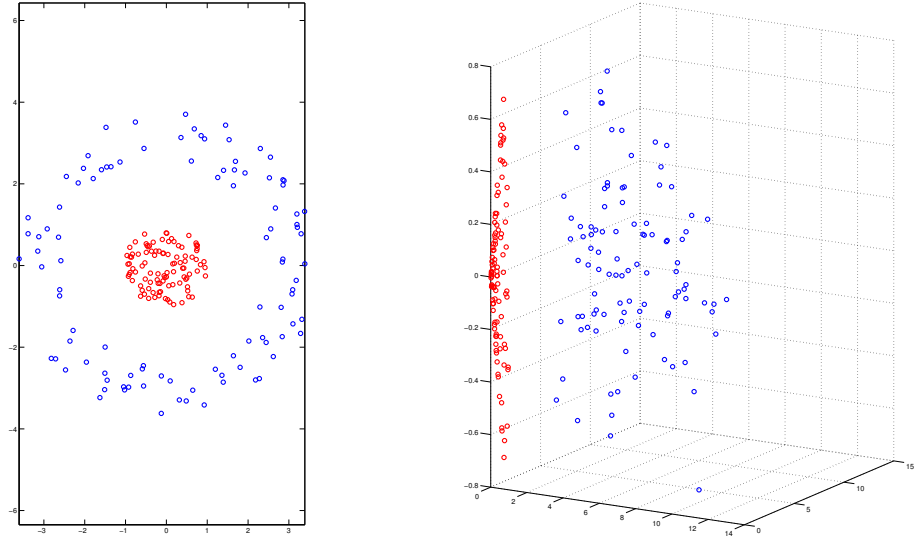


Figure 3.3: Left: Two dimensional toy problem where the background (red box) is a donut around the ball of foreground (blue box). Right: After mapping to the three dimensional feature space where there exists a hyperplane that separates the two data types.

we see that in three-dimensions, Figure 3.3, there exists a hyperplane that separates the foreground and background.

There is a computational cost to working with higher dimensional data. In many real life applications the dimension of the new space that linearly relates the data can be quite large, in some cases of infinite dimension. To mitigate computational complexity, some algorithms can be written to rely on only the inner product between the data points. In this case we can replace  $\langle \Phi(\cdot), \Phi(\cdot) \rangle$  with a *kernel function* defined as

$$\kappa(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle. \quad (3.3)$$

In the toy example pictured we can explicitly find the kernel function

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle [x_1^2, x_2^2, \sqrt{2} x_1 x_2], [z_1^2, z_2^2, \sqrt{2} z_1 z_2] \rangle = (x_1 z_1 + x_2 z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 \quad (3.4)$$

where  $\mathbf{x}$  and  $\mathbf{z} \in \mathcal{X}$ . We call this kernel the *quadratic kernel* or *polynomial kernel* of degree two. Note that the correspondence between kernel and feature map is not unique. For example the map  $\Phi : [x_1, x_2]^T \mapsto [x_1^2, x_2^2, x_1x_2, x_2x_1]^T$  yields the same kernel. Regardless of the dimension of the feature space, we now explain that we only need to explicitly determine the kernel of the feature space, eliminating the computational cost of evaluating the feature map and of applying the linear method to higher order data.

A kernel method is a linear method where the explicit feature space is not required; only the similarity between the data in the feature space is needed. Any linear method that can be written in terms of only the inner product of the data (similarity of data) can be transformed into higher dimensional space by replacing the inner product with the kernel function which implicitly transforms the space into the feature space without requiring the data to be mapped to the higher dimensional space. Many popular data analysis methods can be expressed as kernel methods, including linear regression, principal component analysis (PCA), Fisher discriminant analysis (FDA), and support vector machines (SVM) [77].

We next discuss how to find the kernel *soft-margin* SVM, which we will simply call *Kernel SVM*.

Recall from Chapter 2 that the *soft-margin* SVM is given by

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \|\boldsymbol{\xi}\|_1 \text{ s.t. } u_i(\langle \mathbf{w}, \mathbf{p}_i \rangle - b) \geq 1 - \xi_i, \\ \xi_i \geq 0 \text{ for all } i = 1, \dots, k. \end{aligned} \tag{3.5}$$

where  $\mathbf{p}_i$  denotes the training pixels (background and foreground),  $u_i$  is the classi-

fication of the training pixels,  $\langle \mathbf{w}, \mathbf{p} \rangle - b = 0$  is the hyperplane, and  $\xi_i$  is the slack variable controlled by the penalty parameter  $C$ . We find the Kernel SVM by applying the Karush-Kuhn-Tucker (KKT) condition [35] to find the dual formulation of (A.7) where the dual is a function of the inner product.

The *primal* (A.7) problem has an equivalent *dual* form:

$$\max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j u_i u_j \langle \mathbf{p}_i, \mathbf{p}_j \rangle \quad (3.6)$$

$$\text{s.t. } \sum_i u_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C. \quad (3.7)$$

See Appendix A for the derivation.

The relationship between primal and dual variables is given by

$$\mathbf{w} = \sum_i \alpha_i u_i \mathbf{p}_i. \quad (3.8)$$

The *hard margin* SVM results in the same dual formulation except the dual variable,  $\alpha_i \geq 0$ , is not constrained by  $C$ .

In the dual form of SVM we see that our problem is only dependent on inner products of data vectors. If we use a feature map, we can simply replace  $\langle \mathbf{p}_i, \mathbf{p}_j \rangle$  by  $\langle \Phi(\mathbf{p}_i), \Phi(\mathbf{p}_j) \rangle$  and then by  $\kappa(\mathbf{p}_i, \mathbf{p}_j)$  resulting in the *kernel SVM* dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j u_i u_j \kappa(\mathbf{p}_i, \mathbf{p}_j) \\ \text{s.t. } \sum_i u_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C. \end{aligned} \quad (3.9)$$

Note that (3.9) is a quadratic optimization problem where the objective function can be written as

$$\mathbf{e}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{D} \mathbf{K} \mathbf{D} \boldsymbol{\alpha} \quad (3.10)$$

where  $\mathbf{D} = \text{diag}(u_i)$  and  $\mathbf{K}_{ij} = \kappa(\mathbf{p}_i, \mathbf{p}_j)$ . Problem (3.9) has a unique solution if  $\mathbf{K}$  is symmetric positive semi-definite (and if and only if  $\mathbf{K}$  is conditionally positive definite which we discuss in Section 3.2.2). Therefore we are interested in functions  $\kappa(\cdot, \cdot)$  that guarantee this condition.

## 3.2 Characteristics of Kernels

As we saw in the previous section, a linear method that can be expressed in terms of the inner product of the data can be mapped implicitly to a higher dimensional feature space where we can replace  $\langle \Phi(\mathbf{p}_i), \Phi(\mathbf{p}_j) \rangle$  with  $\kappa(\mathbf{p}_i, \mathbf{p}_j)$ . In order to make this substitution, the map  $\Phi$  needs to be defined on an inner product space

$$\langle \cdot, \cdot \rangle_{\mathcal{H}}. \quad (3.11)$$

For a map  $\Phi$ , if we evaluate the corresponding kernel function over a set of real vectors  $\mathbf{x}_i$  for  $i = 1 \dots n$ , the result will be an  $n \times n$  matrix  $\mathbf{K}$  which is often referred to as the *Gram Matrix* [77]. Each element of the matrix is given by

$$\mathbf{K}_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle. \quad (3.12)$$

Given that  $\Phi$  is defined on an inner product space, we see that  $\mathbf{K}$  is symmetric and positive semi-definite as it is the matrix formed by finding the inner product of the inputs in the feature space. This gives us some indication that a kernel that is a *finitely positive semi-definite function*<sup>1</sup> can be defined as  $\kappa(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$  for

---

<sup>1</sup>This is a kernel function where the matrix formed by evaluating over a finite space is symmetric and positive semi-definite [77].



some map  $\Phi$ .

The following theorem says that, in fact, any finitely positive semi-definite function  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a kernel function.

**Theorem 1** (Characteristic of a Kernel [77]). *A function  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  can be expressed as*

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$$

*with a space  $\mathcal{H}$  and map  $\Phi : \mathcal{X} \rightarrow \mathcal{H}$  for all  $\mathbf{x}, \mathbf{z} \in \mathcal{X}$  if and only if  $\kappa$  is finitely positive semi-definite.*

### 3.2.1 Constructing Kernels

In the previous section we saw that any finitely positive semi-definite function can be a kernel function. Given any kernel, for example,  $\kappa(x, z) = \langle \mathbf{x}, \mathbf{z} \rangle$ , we can build upon it to create more complex kernels [77].

The following functions are kernels:

- $\kappa(\mathbf{x}, \mathbf{z}) = a\kappa_1(\mathbf{x}, \mathbf{z})$  where  $a > 0$  and  $\kappa_1(\mathbf{x}, \mathbf{z})$  is a kernel.
- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z}) + c$  where  $c > 0$  and  $\kappa_1(\mathbf{x}, \mathbf{z})$  is a kernel.
- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z}) + \kappa_2(\mathbf{x}, \mathbf{z})$  where  $\kappa_1(\mathbf{x}, \mathbf{z})$  and  $\kappa_2(\mathbf{x}, \mathbf{z})$  are kernels.
- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z})\kappa_2(\mathbf{x}, \mathbf{z})$  where  $\kappa_1(\mathbf{x}, \mathbf{z})$  and  $\kappa_2(\mathbf{x}, \mathbf{z})$  are kernels.

Each one of these properties can be verified by constructing the kernel matrix and confirming that it is positive semi-definite ( $\mathbf{v}'\mathbf{K}\mathbf{v} \geq 0$  for all vectors  $\mathbf{v}$ ).

### 3.2.1.1 Polynomial kernels

The quadratic kernel from the toy problem is an example of a polynomial kernel of degree two. The most general form is given by

$$\kappa(\mathbf{x}, \mathbf{z}) = (a\langle \mathbf{x}, \mathbf{z} \rangle + c)^d \text{ for } a, c \text{ and } d > 0. \quad (3.13)$$

This kernel has three parameters ( $a$ ,  $c$ , and  $d$ ). Often when used with Kernel SVM over normalized data,  $a = 1$  and  $c = 0$ , which is called the homogeneous polynomial kernel.

### 3.2.1.2 Gaussian Kernels

The inner product measures the similarity of two data points but in many applications the measure of dissimilarity between two data points may be more useful. The dissimilarity is measured by distance. The Gaussian kernel is given by

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \text{ for } \sigma > 0, \quad (3.14)$$

and is a measure of dissimilarity.

The Gaussian, often referred to as the Gaussian Radial Basis Function or RBF [76], is a finitely positive semi-definite kernel. The feature space is of infinite dimension which can be seen if we generalize the Gaussian kernel function to a polynomial kernel function by Taylor expansion.

The Gaussian kernel is dependent on the parameter  $\sigma$ . For  $\sigma$  approaching zero the kernel becomes the identity matrix, where each data point of the training data

is fit exactly. For  $\sigma$  approaching infinity, the kernel matrix approaches a constant matrix, losing all information of similarity among data points.

The Gaussian kernel is based on a measure of dissimilarity. A larger class of kernels based on dissimilarity which are not guaranteed to be finitely positive semi-definite, but are conditionally positive definite, follows in Section 3.2.2.

### 3.2.2 Conditionally Positive Definite Kernels

A conditionally positive definite matrix is defined as

**Definition 1.** [*Conditionally Positive Definite Matrix*] A symmetric  $\mathbf{K}$  is conditionally positive semi-definite (CPD) if

$$\sum_{i,j=1}^k v_i v_j \mathbf{K}_{ij} \geq 0 \text{ for all } \mathbf{v} \text{ such that } \sum_{i=1}^k v_i = 0. \quad (3.15)$$

Details on conditionally positive definite matrices can be found in [2]. A kernel is CPD if its Gram matrix for any choice of real vectors is CPD.

The kernel  $\kappa(\mathbf{x}, \mathbf{z}) = -\|\mathbf{x} - \mathbf{z}\|^2$  is an example of a CPD kernel [4]. We can see this by using the identity

$$-\|\mathbf{x} - \mathbf{z}\|^2 = -\|\mathbf{x}\|^2 - \|\mathbf{z}\|^2 + 2\langle \mathbf{x}, \mathbf{z} \rangle. \quad (3.16)$$

Then

$$\sum_i \sum_j v_i v_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = -\sum_i \sum_j v_i v_j \|\mathbf{x}_i\|^2 - \sum_i \sum_j v_i v_j \|\mathbf{x}_j\|^2 + 2 \sum_i \sum_j v_i v_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad (3.17)$$

and

$$2 \sum_i \sum_j v_i v_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \geq 0 \quad (3.18)$$

when  $\sum_i v_i = 0$ .

As suggested by the example, there is a relationship between measures of dissimilarity and conditionally positive definite kernels. This relationship is expressed by the identity

$$-\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|^2 = -\kappa(\mathbf{x}, \mathbf{x}) - \kappa(\mathbf{z}, \mathbf{z}) + 2\kappa(\mathbf{x}, \mathbf{z}), \quad (3.19)$$

where  $-\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|^2$  is a CPD kernel.

In the case of kernel SVM given by (3.9) we see that it has a unique solution for all CPD kernels. By plugging (3.19) into (3.9) we find that the terms corresponding to  $\kappa(\mathbf{x}, \mathbf{x})$  and  $\kappa(\mathbf{z}, \mathbf{z})$  go to zero because of the SVM constraint.

In practice, CPD kernels have been shown to be very effective for Kernel SVM [4, 80]. More details on CPD kernels are found in [4, 75, 76].

### 3.2.2.1 Log Kernel

The log kernel is an example of a CPD kernel which has been suggested for image processing applications [80]. It is given by

$$\kappa(\mathbf{x}, \mathbf{z}) = \log(\|\mathbf{x} - \mathbf{z}\|^d + 1) \text{ for } 0 < d \leq 2. \quad (3.20)$$

The log kernel has been suggested for color image recognition problems, particularly *class-vs-all other classes* [4] as well as spectral reflectance problems [25].

### 3.2.3 Other Notable Kernels

The *Multilayer Perceptron (MLP)* is given by

$$\kappa(\mathbf{x}, \mathbf{z}) = \tanh(a\langle \mathbf{x}, \mathbf{z} \rangle + c). \quad (3.21)$$

This kernel is also known as the *Hyperbolic Tangent Kernel* or *Sigmoid Kernel*, developed originally as a model for Neural Networks. The origin of SVM is often attributed to the field of network theory [80].

A well-cited list of kernel functions used for machine learning, including conditionally positive definite kernels, can be found in [80].

### 3.3 Kernel and Kernel Parameter Selection

Finding the appropriate kernel for a particular task and method is not trivial. Poor kernel and parameter choice can lead to poorer classification of data. Both the choice of the kernel and kernel parameter are dependent on the data as well as the particular kernel method used. Practitioners use past experience, knowledge of the particular application or an exhaustive search over a subset of kernels and parameters to find an appropriate pair for a particular task. Because of the possible bias that may be introduced and the cost of an exhaustive search, many have considered automated processes for picking kernels and parameters.

#### 3.3.1 Selecting a Kernel

Understanding the characteristics of the task can help us find a kernel that captures the measure of dissimilarity/similarity for the particular task and data. Unfortunately such intuition about data is rare.

Several authors have introduced automated kernel selection methods [7, 45, 67, 79]. Their general approach is to build a space of kernels and optimize the

kernel method over the kernel space. These methods have been coined Multi-Kernel Learning or MKL methods by [79].

Generally MKL methods assume that the subset of kernels can be represented as a composite kernel

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{K=1}^M d_K \kappa_K(\mathbf{x}, \mathbf{z}), \quad (3.22)$$

where  $M$  is the number of kernels and  $\kappa_K(\mathbf{x}, \mathbf{z})$  is a given kernel. Once a kernel space is built several approaches have been suggested for searching over that space to find the optimal kernel.

The authors of [79] constructed a convex set of kernels and added a quadratic constraint over the set of kernels into the Kernel SVM optimization problem. They showed that the MKL SVM is a semi-infinite linear program which can then be solved in a number of different ways. They suggested a two step method alternating between the standard SVM problem and a generic cost function with the additional constraints on the weights of the composite kernel. Similar to this method, [45] suggested a method where the Kernel SVM is written as a semi-definite program, again with constraints over the set of composite kernels which can also be solved iteratively in a two step approach.

In [45], the authors found that their MKL algorithm, over a composite kernel made up of three different kernels (polynomial kernel of degree 2, Gaussian kernel with  $\sigma = .5$  and linear kernel), had minimal improvement, if any, over Kernel SVM with a single Gaussian kernel where the parameters were tuned. Because of the limited benefit and additional computational cost of the MKL methods, they were

not included in our study.

### 3.3.2 Parameter Selection

Finding the kernel parameters for Kernel SVM is often done in conjunction with tuning the penalty parameter  $C$  by cross validation over a grid [17, 76]. This approach is not ideal as it can take time and the range of optimal parameters can be small. In [14] and [85], they suggest that the appropriate kernel and kernel parameter can be found using application-domain knowledge by an expert user. For example they suggest that the Gaussian parameter  $\sigma$  should reflect the distribution of the training data.

Methods for automatically finding the kernel parameter have received some attention. In [15], they suggest a gradient descent algorithm over several model selection criteria. Using a gradient descent algorithm they found similar results, compared to searching over a grid, in significantly less time [15].

In our case we are limiting ourselves to kernels that are dependent on a single parameter, and we used default parameters for testing. For kernels with many parameters these selection methods become more appealing.

## 3.4 Kernel SVM as a Domain-Specific Grayscale Method

Similar to the case of the linear SVM, we define the DSGS image as

$$y_j = \langle \Phi(x_j), \mathbf{w} \rangle - \mathbf{b}, \quad (3.23)$$

where the value of  $y$  represents the distance from the hyperplane in the feature space.

If we plug in the relationship between the primal variable and dual variables, given by  $\mathbf{w} = \sum_i \alpha_i u_i \Phi(\mathbf{p}_i)$ , then the distance from  $x_j$  to the hyperplane in feature space is given in terms of the kernel function by

$$y_j = \sum_i \alpha_i u_i \kappa(\mathbf{x}_j, \mathbf{p}_i) - b. \quad (3.24)$$

Given our understanding of the mechanics of SVM, we know that the data that are near the hyperplane have a greater probability of being misclassified when compared to points far from the hyperplane. There has been work to explicitly convert the resulting SVM score to a conditional probability  $P(Y = \pm 1 | x_j) = \sigma(y_j)$  but this requires additional training data [65, 71].

### 3.5 Non-Kernel Nonlinear Domain-Specific Grayscale Method

In this section we introduce a new nonlinear domain-specific grayscale method where we learn the two distinct three-dimensional regions that enclose the background and foreground pixels in **RGB** space. We call these inclosed regions the *background gamut* and *foreground gamut*. The motivation for explicitly learning the gamuts comes from the fact that choosing the right kernel and paired parameters is a difficult and time consuming task.

Color images are generally only three-coordinates, therefore it is feasible to try to learn the background and foreground gamuts from the data. Once the gamuts are learned then we can classify other parts in the image by determining which region



they should be members. If the background and foreground are made up of several different colors the resulting regions are not guaranteed to be convex (as shown in Figure 3.2). Learning non-convex shapes is not a trivial problem.

Similar problems arise in three dimensional surface reconstruction which has applications in computer graphics. In these applications there is additional information known about the collection of points [8]. Generally these point-clouds have an additional dimension or dimensions, such as an intensity or a normal vector giving the reference in which the data was captured. In our case we only have the location of the pixel in three dimensional **RGB** space and because our shapes may not be convex we cannot use methods such as triangulation to learn these shapes.

Consider just the foreground. If we assume that each data point is a heat source and then let the the heat diffuse over a fixed time, we can define a normalized temperature as

$$\Psi(\mathbf{x}) = \sum_{i \in P} \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}), \quad (3.25)$$

where  $\sigma$  is a diffusion parameter. Then, for some level  $L$ , the set of points with that temperature is given by

$$\Gamma_L = \{\mathbf{x} \mid \Psi(\mathbf{x}) = L\}. \quad (3.26)$$

If we want to separate the background and foreground, we assume that the background data points are heat sinks and are given by the negative of the same process. Then our temperature is

$$\Psi(\mathbf{x}) = \sum_{i \in P \cup \bar{P}} u_i \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}), \quad (3.27)$$

and the level surface is

$$\Gamma_L = \{(\mathbf{x}) | \Psi(\mathbf{x}) = L\}. \quad (3.28)$$

We call this method the Gaussian Sum (GSUM) method.

One might assume that the zero level,  $\Gamma_0$ , would be the best boundary between the shapes. Unfortunately the foreground and background pixels are likely to have different densities and number of pixels. Consider the contour plot for a two-dimensional toy problem, seen in Figure 3.4. In this case the boundary found at  $L = 0$  (blue line) is clearly not the best boundary.  $L = 2$  (teal line) is a more reasonable boundary.

To partially account for differences in the number of pixels and potential differences in density,  $\sigma_{GSUM}$  was rescaled by  $n/(2n_1)$  for the foreground and  $n/(2n_2)$  for the background where  $n_1$  is the number of elements in the foreground and  $n_2$  is the number of elements in the background and  $n = n_1 + n_2$ . In addition to determining the level  $L$ , we used the thresholding methods presented in Chapter 2.

Although the motivations for the GSUM method and the Gaussian Kernel SVM are different, they result in a similar expression of the DSGS image. The DSGS image for the Gaussian Kernel SVM is given by

$$y_j = \sum_i \alpha_i u_i \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{p}_i\|^2}{2\sigma^2}\right) - b. \quad (3.29)$$

In Chapter 6 we discuss further work on improving upon the GSUM method as well as other methods for learning the foreground and background gamuts that are less reliant on user-determined parameters.

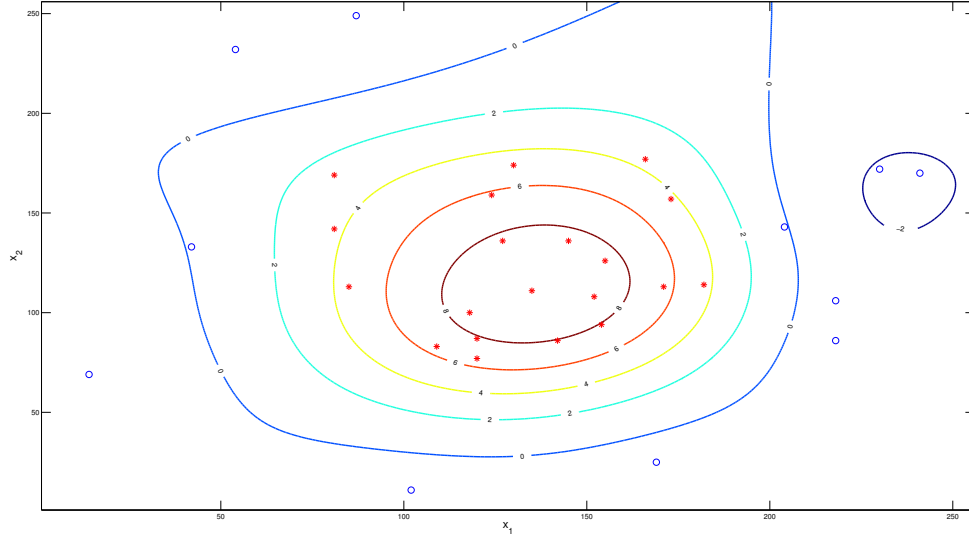


Figure 3.4: Level sets for two-dimensional pixels. Contour plot of  $\Psi(\mathbf{x}, u)$  where the red stars are the foreground pixel and blue dots are the background. The dark blue curve is the zero level set.

### 3.6 Results with the Non-Linear Domain-Specific Grayscale Methods

In this section, we present results comparing the nonlinear DSGS methods and discuss some advantages of nonlinear methods over the linear DSGS methods.

For this work we focused our efforts on three different kernels. The Gaussian and polynomial kernels were used because of their wide-spread use in the literature. The log kernel was chosen because of its suggested use in related problems (color image recognition problem [4] and spectral reflectance [25]). We name the Kernel SVM implementation for each of the kernels as follows:

- Gaussian Kernel SVM (GSVM),
- Polynomial Kernel SVM (PSVM),

- Log Kernel SVM (LSVM).

For each of these kernels we use the default SVM parameter  $C = 1$  and kernel parameters:

- GSVM:  $\sigma = 1$ ,
- PSVM:  $p = 2$ ,
- LSVM:  $d = 1$ ,

unless otherwise stated.

### 3.6.1 Synthetic Image Testing

Table 3.1 shows the percent misclassified after thresholding for the synthetic images for each of the nonlinear methods as well as for the MAD method. The MAD method was chosen as the benchmark linear method because it is not reliant on user-determined parameters.

GSUM produced the best segmentation for the “*Nonlinear Dot*” and reasonable segmentation for the “*Linear Dot*,” although for the images based on data from natural images, the GSUM method generally failed to produce reasonable segmentations. For the synthetic images, the margins are well defined and the density between samples are constructed to be similar. For the natural images, the margin is not as well defined and there can be significant differences in densities between the training background and foreground which creates a lot of issues for GSUM. Some issues can be mitigated by tuning the parameter  $\sigma_{GSUM}$  as well as improving the rescaling of  $\sigma_{GSUM}$ .

As expected, all the kernel methods worked well for the “*Nonlinear Dot*” but for most of the other synthetic images they produced little or no improvement over the linear methods. Images that are made up of primarily two distinct colors, such as the “*Linear Dot*,” “*Football Dot*,” and “*Horse Dot*,” suggest a linear solution. Considering only the synthetic images, LSVM was the most consistent, producing the best segmentation for five of the images and for the remaining seven produced segmentations that are within range of the best segmentations. In contrast, GSVM produced a significantly worse segmentation for the image “*Football Dot*.” PSVM produced a significantly worse segmentation for the image “*Saturn Dot*.”

Measuring the effectiveness of a method purely on the percent misclassified can be misleading. If we examine the histograms of the DSGS images as seen in Figures 3.5 - 3.6, we see that even if the image is segmented correctly there is variation between methods. In the example of “*Owl Dot*” each of the Kernel SVM methods improves upon the linear method, although, as Figure 3.6 shows, there remains significant overlap between the background and foreground. In Figure 3.5, compared to the results in Figure 2.20, we see that although each of the methods classified all the pixels correctly, the background is better clustered for the nonlinear SVM methods, resulting in a DSGS image with a more uniform background. We discuss this idea further in Section 3.7.

Table 3.1: Percent misclassified for the MAD method and the nonlinear methods on the synthetic images.

<b>Image</b>	<b>MAD</b>	<b>GSVM</b> $\sigma = 1$	<b>PSVM</b> $p = 2$	<b>LSVM</b> $d = 1$	<b>GSUM</b> $\sigma = 100$
Linear Dot	0	0	0	0	2.81
Non linear Dot	44.65	5.47	5.52	3.54	3.47
Skin Dot	0	0.71	0.12	0.83	13.45
Pepper Dot	0	0	0	0	45.02
Football Dot	0	12.55	0	0.15	6.84
Lions Dot	0.05	0.22	0.32	0.37	55.54
Saturn Dot	0.02	0.05	6.18	0.02	4.88
Lions 2 Dot	2.88	2.42	4.76	2.29	35.82
Owl Dot	16.53	12.40	15.63	11.33	28.86
Tiger Dot	15.92	15.87	13.31	14.48	63.13
Goat Dot	17.85	28.91	26.83	29.42	23.85
Horse Dot	0	0.71	0	0.71	10.91

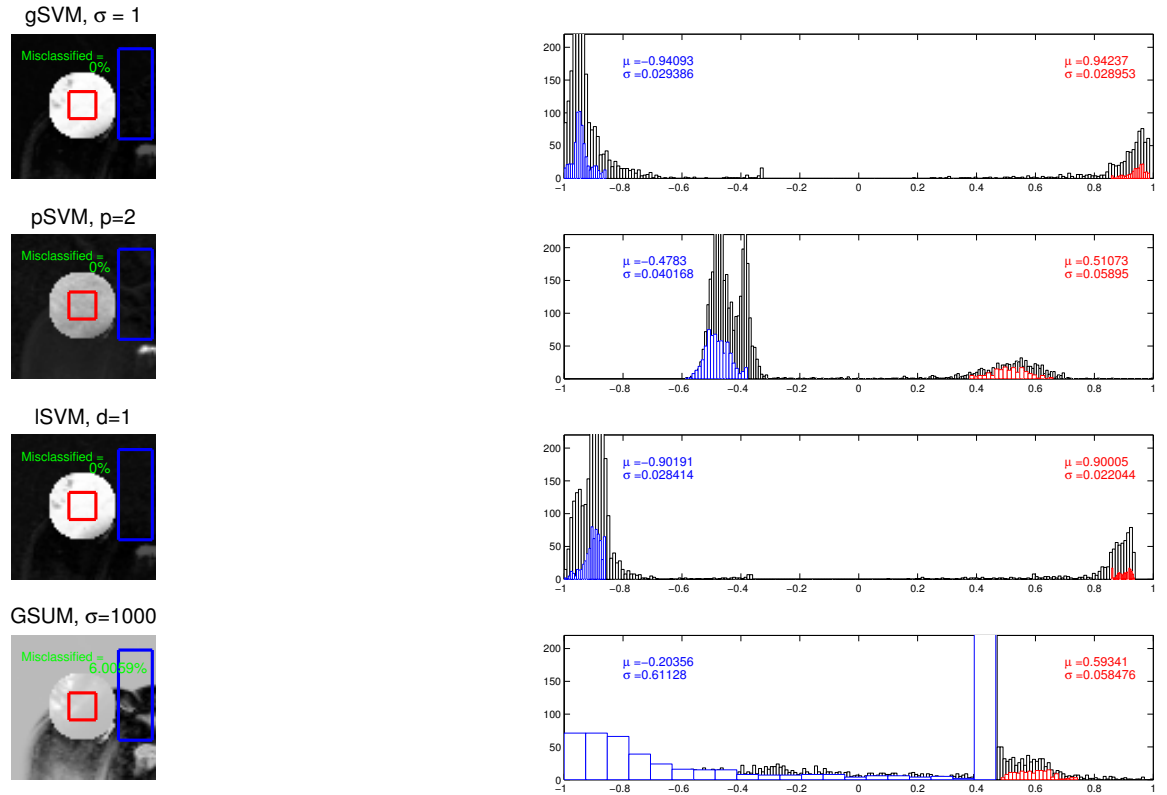


Figure 3.5: Results of the nonlinear DSGS methods on the “*Pepper Dot*” image. In the histograms on the right, pixels of  $P_w$  and  $\bar{P}_w$  are shown in red and blue respectively.

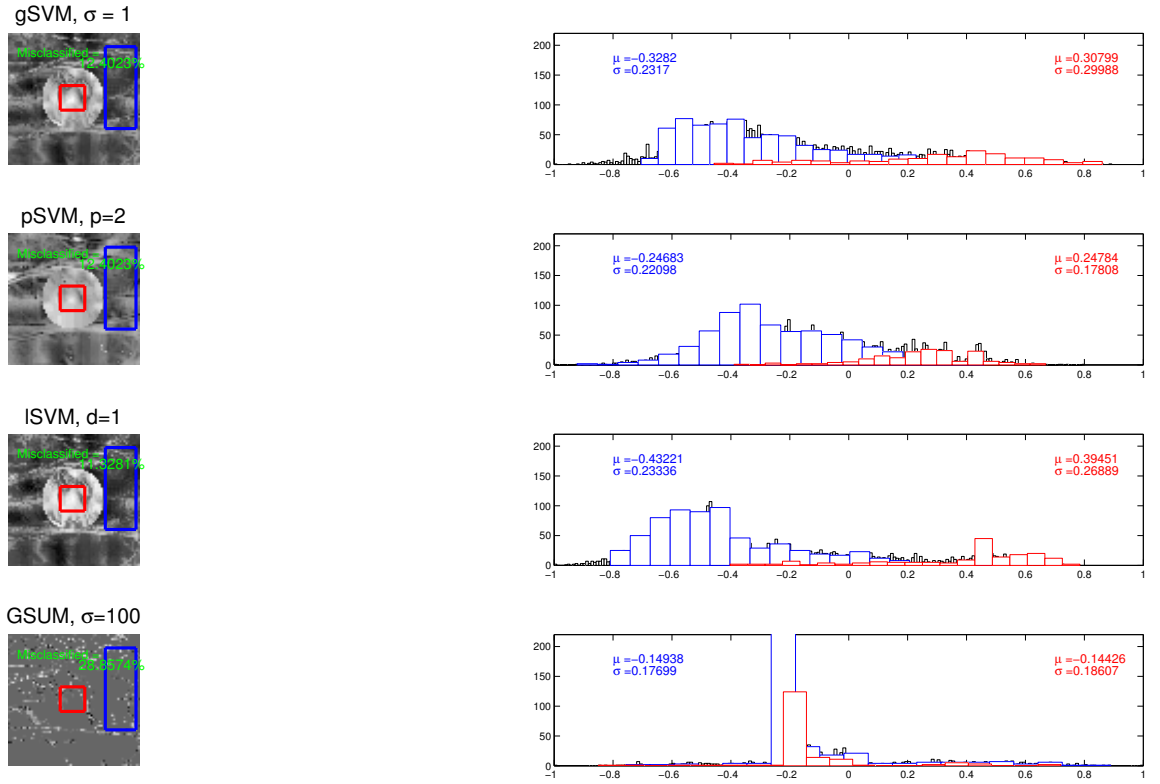


Figure 3.6: Results of the nonlinear DSGS methods on the “Owl Dot” image. In the histograms on the right, pixels of  $P_w$  and  $\bar{P}_w$  are shown in red and blue respectively.



### 3.6.2 Results on Natural Images

The linear MAD DSGS method generally produced better segmentations over the nonlinear methods as seen in Table 3.2. As previously observed, simply segmenting the image does not give us the whole picture.

In Figure 3.7, of the Owl, none of the methods clearly differentiate between the foreground and background. Though the foreground and background are made up of a number of distinct colors, one of the colors in the foreground is very similar to a color in the background. In this case, as seen in the Table 3.2 and in Figure 3.7, LSVM finds the best DSGSs for segmentation, significantly capturing more of the body of the owl.

In the example of the image of lions, as seen in Figure 3.8, we show how dependent nonlinear methods are on the training information. The left column shows the results for smaller background and on the right we show results for an expanded selection of background pixels. In both cases we find that for the MAD method more of the pixels within the body are misclassified, particularly around the noses of the lions, but the background is correctly classified. The ability of the linear method to correctly classify the background suggests that the variations in the color of the background are somewhat linearly related to each other. For the nonlinear methods, the bodies of the lions are better classified in both cases but there was a significant difference in the accuracy of the classification of the background. We find that when selecting a background sample that includes the shadow of the lions and areas that are out of focus (top and bottom of the image) the percent misclassified

Table 3.2: Percent misclassified for the MAD method and the nonlinear methods on the Berkeley Segmentation image dataset.

Image	MAD	GSVM $\sigma = 1$	PSVM $p = 2$	LSVM $d = 1$	GSUM $\sigma = 4000$
Lions	6.25	8.47	13.18	10.7	51.01
Lions 2	3.27	3.34	6.40	4.27	21.13
Owl	22.67	21.40	21.80	20.32	52.31
Tiger	31.00	60.17	58.67	60.55	40.84
Goat	25.63	20.63	35.62	22.58	32.56
Horses	2.17	2.42	2.89	2.52	12.64

improves but additional training pixels come at a significant computational cost. In Table 3.3 we show the percent misclassified for each of the nonlinear DSGS methods and the linear DSGS method along with the time to construct the image in seconds. For example, although the GSVM method produced the best classification when a larger background was selected, it took 40 times longer to compute. In general we found that, for the cases with large training samples coupled with no clear margins, the SVM based DSGS methods were time consuming both to train and in constructing the image.

### 3.7 Advantages and Disadvantages of Kernel SVM DSGS Methods

The major advantage of the Kernel SVM DSGS is the ability to cluster areas made up of many distinct colors. Even in cases where the linear methods outperformed in segmenting (classifying) the image, the Kernel SVM DSGS method resulting image may be better suited for the task. This is demonstrated in Figure 3.8 where GSVM and LSVM produce more accurate edges around the lions and

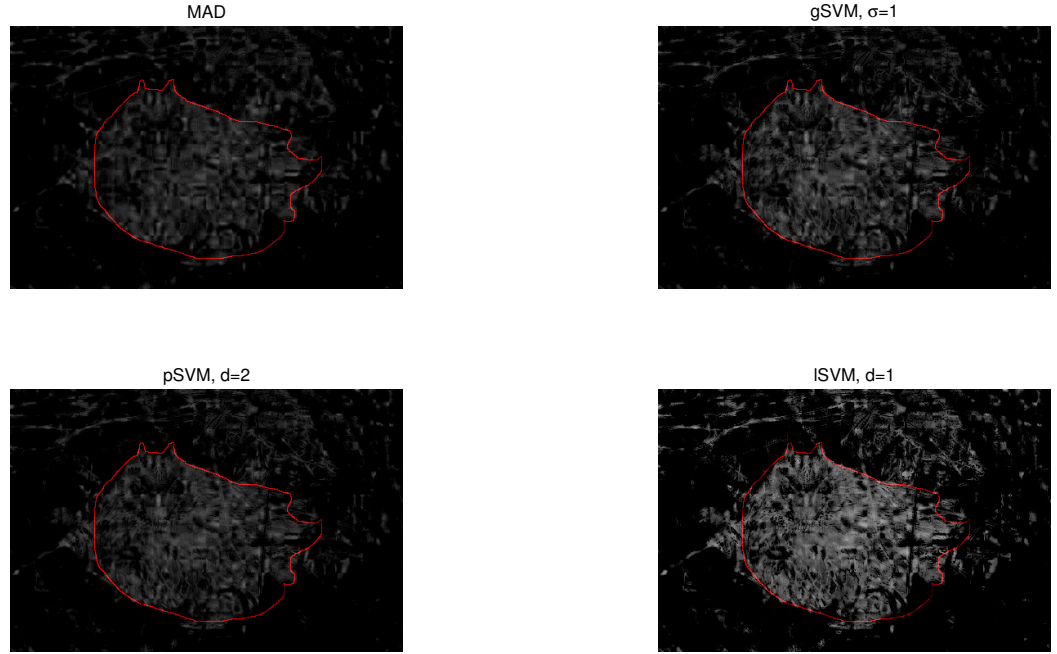
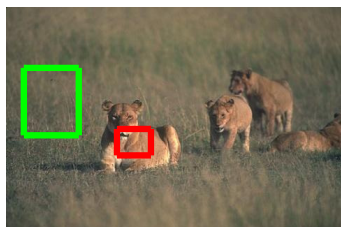


Figure 3.7: Resulting DSGS image for MAD, gSVM, pSVM and lSVM for the image of an Owl with the human-determined boundary superimposed in red. The images are displayed so that values less than zero are set to black to highlight the classification of the foreground.

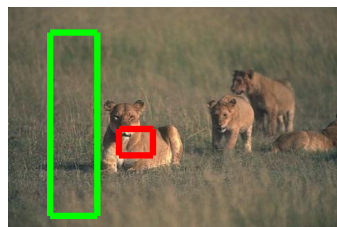
Table 3.3: Percent misclassified and training time for the MAD method and nonlinear SVM methods for the image *Lions*.

Image	MAD	GSVM $\sigma = 1$	PSVM $p = 2$	LSVM $d = 1$
Background 1				
% Missclassified	6.25	8.47	13.18	10.70
Time (s)	1.15	5.23	1.73	7.67
Background 2				
% Missclassified	6.30	5.23	11.78	5.94
Time (s)	1.27	119.74	1.77	14.46

#105053 – Lions



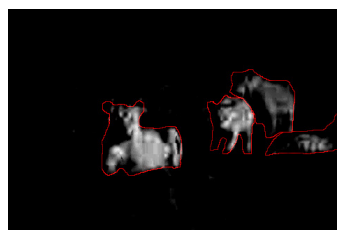
#105053 – Lions



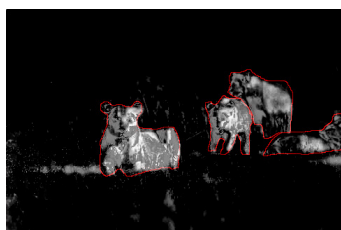
MAD



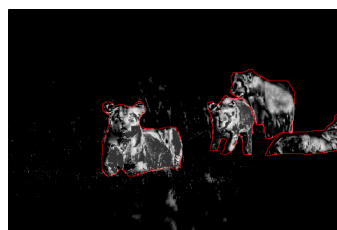
MAD



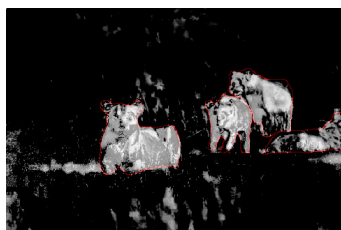
gSVM,  $\sigma=1$



gSVM,  $\sigma=1$



ISVM,  $d=1$



ISVM,  $d=1$

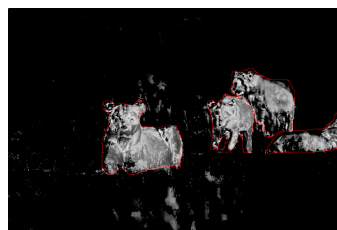


Figure 3.8: Results for two different selected backgrounds of significantly different sizes. The images are displayed so that values less than zero are set to black to highlight the classification of the foreground.

the misclassified pixels in the background may be easily discarded because of the locality. This suggests the importance of finding a feature space that not only separates the foreground but also clusters the foreground pixels within their class. This becomes particularly evident if we consider the color image of M&Ms, top left of Figure 3.9. In this case the user wants to segment a single color M&M from the rest of the image. The DSGS image for the MAD linear method and the three Kernel SVM methods are seen in Figure 3.9. In that figure we see that the nonlinear methods best cluster the six colors (yellow, orange, blue, brown and green M&Ms and the color of the table) distinct from the red M&Ms. Clearly, the DSGS images for the nonlinear methods much better differentiate the red M&Ms from the rest of the image, making the task of segmenting the red M&Ms easier.

Depending on the task, the advantages of the Kernel SVM could be crucial to finding a DSGS that makes the task as easy as possible. The advantages are dependent on many components including the choice of kernel, the kernel parameter, the background and foreground selection, and characteristics of the image. These in turn, can be very sensitive to variations. In addition to being dependent on many components, these methods can be time consuming depending on the number of support vectors found by solving the optimization problem (nonzero dual variables). The number of support vectors is not necessarily dependent on the number of training pixels but rather it is dependent on the complexity of the margins. For example, for the image of the Owl, Tiger and Goat, the kernel SVM DSGS methods took significantly longer to compute than for example the image of the Horses (several minutes versus seconds).

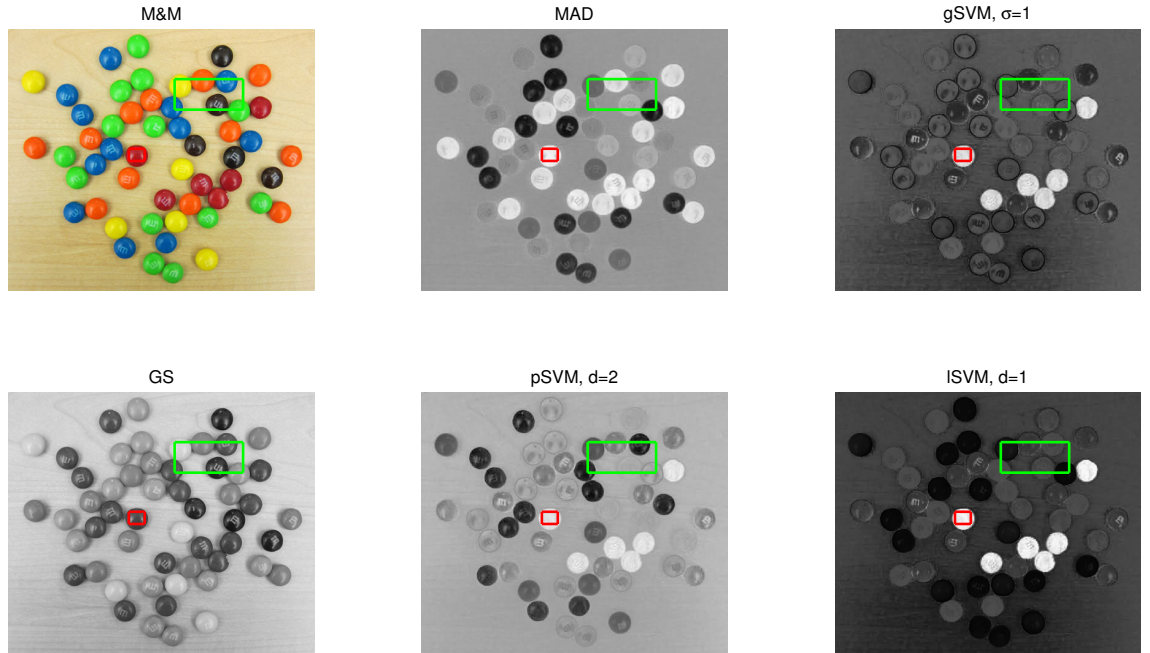


Figure 3.9: Color image of M&Ms and resulting DSGS images for the MAD, gSVM, GS, pSVM and lSVM.

### 3.8 Summary and Conclusion

In this chapter we used Kernel SVM as a nonlinear DSGS method and introduced a new nonlinear DSGS method that attempts to learn the relationships between the background and foreground. We discussed several kernels as well as how to choose possible kernels for a given task. For this work we limited our study to three different kernels chosen either because of their popularity across many different applications or because of their suggested use for similar problems.

In testing on the synthetic images and images from the Berkeley segmentation dataset, the linear methods often produce the best segmentation as measured by percent misclassified. Even when the linear method out-performed the nonlinear

methods, as measured by percent misclassified, the DSGS image for the nonlinear method may be a better choice depending on the task at hand. As we saw in Section 3.7 the nonlinear Kernel SVM can cluster backgrounds and/or foregrounds with a lot of variation in color but the quality of this transformation is very sensitive to the user selections.

For cases where the background and foreground are each primarily made of a single color and distinct from one another, linear methods should be used. For all other cases, and depending on the task at hand, further investigation should be used to determine if a nonlinear method could improve the results. For example, viewing the scatter plot of the training samples before transformation may give us clues as to whether a linear or non-linear method should be used. After finding the DSGS image, the histogram may give one insights into how useful the results are for a particular task. If the task calls for a nonlinear DSGS method, we found that the LSVM method produced the most consistent results and was least dependent on the parameters for natural images.

In Chapter 4 we apply both the linear and nonlinear DSGS methods to analysis of digital images in dermatology.

## Chapter 4: Using Domain-Specific Grayscale Method for Dermatology Digital Image Analysis

Dermatology is a branch of medicine that diagnoses and treats diseases of the skin, hair and nails. In the practice of dermatology, traditionally the clinicians rely on visual inspection and palpation to help in diagnosing different skin conditions, for example in diagnosing skin cancer [51] or monitoring the reddening after skin irritation [60]. Noninvasive digital visualization techniques are becoming essential tools in care delivery [6], but great variation exists between experienced and inexperienced dermatologists in regards to the success rate for diagnosing skin cancer [51] using these techniques.

In this chapter we apply our techniques for finding a Domain-Specific Grayscale (DSGS) image to improve the contrast between reference skin and lesions, *erythema* (reddening), or *wheals* (raised hives) for improved visual inspection and computer-aided tasks. Real-world applications include monitoring the spread of infection, measuring the effectiveness of the treatment of a skin condition, determining positive allergic reactions after skin-prick testing, and detecting or monitoring skin cancers. In Section 4.1 we discuss the current use of digital imaging in the field of dermatology and some existing computer-aided diagnostic systems. In Section 4.2 we discuss



measuring area as well as changes in intensity of erythema and wheals after skin-prick testing and present some results of using the DSGS methods and how the resulting image can aid in the task of detecting and measuring the affected skin. In Section 4.3 we discuss applications for detecting and determining skin lesion boundaries, particularly as it relates to skin cancers such a melanoma. Lastly in Section 4.4 we summarize the impact of our new DSGS methods in dermatology and discuss future work for this application.

## 4.1 Imaging-Aided Diagnosis in Dermatology

Dermatologists have long used photography, but since the availability of digital photography the costs are lower. Digital images require no time for developing and can easily be passed between clinicians and pathologists [6]. Dermatologists use digital images for a variety of purposes including during surgery to limit the amount of healthy skin removed, monitoring subtle changes during laser treatment, and as a diagnostic tool for skin cancers.

In addition to commercial digital cameras there are several imaging technologies used by dermatologists to aid diagnosis. Dermoscopy is a method that involves looking at a lesion under illumination and magnification, traditionally with a layer of oil or alcohol added to the skin to decrease uneven illumination. New instruments often use polarized light in place of or in addition to a liquid interface. This method has been developed into a single handheld instrument, often called a dermoscope, used for visual inspection or as a lens to a camera [68]. With an expert

dermatologist the use of dermatoscopes has improved the diagnosis of melanoma, a deadly skin cancer, to about 90% sensitivity (59% specificity), but in the hands of an inexperienced practitioner a rate of only 62% sensitivity is obtained [51].

In the application of skin cancer diagnoses, there are many new technologies being developed that are computer augmented including the technologies MoleMax [54] and Melafind [52]. Imaging techniques such as multispectral imaging, optical coherence tomography, ultrasound, and electrical bioimpedance are often used in these systems. In addition to different types of imaging techniques, many of these new technologies have diagnostic abilities [51]. For example Melafind (currently premarket) captures multispectral images over a range of wavelengths and sends the images to a supercomputer where the images are compared against a 20,000 image database [52].

#### 4.1.1 Preprocessing Images

The goal of preprocessing is to find an image that best differentiates between affected skin and healthy skin for both visual inspection and to aid segmentation, feature extraction, and classification. This can be a difficult task as the transition between healthy and affected skin can be gradual and the healthy skin can have large variations in color and textures. Artifacts such as hair and blood vessels can block or interfere with boundary detection, and illumination can be uneven [51]. Processes such as smoothing, artifact removal through inpainting, color space transformation including to grayscale, and “contrast enhancements” are routinely referenced as

preprocessing techniques used in the literature. However, details are often omitted. In the work of [51], they stated that only 45% of the literature included details of how image were processed in the application of computer aided diagnosis of skin cancer.

#### 4.1.2 Multilayered Skin Model of Absorbance

To help detect and quantify erythema as well as changes in pigmentation, [91] derived a measure for erythema, Erythema Index (EI), and for pigmentation, Melanin Index (MI).

The multilayered skin model for absorbance of the skin at a particular wavelength  $\lambda$  is modeled as

$$A_\lambda = M_\lambda C_m + H_\lambda C_h + D \quad (4.1)$$

where  $A_\lambda$  is the absorbance of the skin,  $M_\lambda$  is a coefficient that depends on the absorbance of the melanin,  $H_\lambda$  is the coefficient for the absorbance of the hemoglobin,  $C_m$  is the coefficient for the amount of melanin in the melanin layer,  $C_h$  is the amount of hemoglobin in the blood layer, and  $D$  is a constant for the “apparent (pseudo) absorbance” of the dermis.

Using *Lambert-Beer’s law*, which is the governing equation relating the absorption and the reflectance of light of a material, we have

$$A_\lambda = \log(1/R_\lambda), \quad (4.2)$$

where  $R_\lambda$  is the reflectance of the skin.

To find the Erythema Index we consider the difference between the model of

absorption at two different wavelengths or

$$A_1 - A_2 = (M_1 - M_2)C_m + (H_1 - H_2)C_h. \quad (4.3)$$

Then if we find the wavelengths where  $M_1 - M_2$  is close to zero, the model can be approximated as

$$A_1 - A_2 = (H_1 - H_2)C_h \quad (4.4)$$

which is solely dependent on the absorbance of hemoglobin (which relates to the cause of erythema). This is the case when the wavelength  $\lambda_1$  ranges from  $540nm$  to  $570nm$  (green) and  $\lambda_2$  is  $660nm$  (red). This leads to the model for the EI index:

$$A_1 - A_2 = \log(1/R_1) - \log(1/R_2). \quad (4.5)$$

The EI index can then be approximated using the **RGB** color space of the image, where  $R_1$  can be approximated by the **G** coordinate and  $R_2$  can be approximated by the **R** coordinate of **RGB**. Then the EI index is given by

$$EI = \log(\mathbf{R}) - \log(\mathbf{G}). \quad (4.6)$$

For the Melanin Index we want to find wavelengths where  $H_1 - H_2$  is close to zero. This is the case when  $\lambda_1$  ranges from  $620nm$  to  $650nm$  and  $\lambda_2$  ranges from  $670nm$  to  $700nm$ . Both  $\lambda_1$  and  $\lambda_2$  are in the spectrum that the **R** coordinate captures so it is not feasible to find an approximate model using the **RGB** image, although if we had access to multispectral images this would be possible. The authors [91] suggest using

$$MI = -\log(\mathbf{R}) \quad (4.7)$$

as most of the light absorbed at the wavelengths that the **R** filter captures is said to be attributed to the melanin.

### 4.1.3 Boundary Detection

Finding the boundary of the affected skin is often the first step in any computer aided diagnostic system. A number of methods for segmenting have been suggested in the literature including different thresholding techniques, gradient edge-detection based techniques, and level-set based techniques.

For this work, since we are looking for closed boundaries of affected patches of skin, we will use level set methods. Applications of level set methods are diverse [84] and have been used successfully in image and video processing. Basics of level set methods can be found in Appendix B. We will use the *Chan-Vese* level set method [13] as it requires no smoothing and assumes that the image is made up of two distinct regions. This is a desirable feature for these applications because many of the boundaries may not be well-defined and smoothing will increase the difficulty of finding faint edges. The numerical implementation used is based on the implicit scheme by [13] implemented by [90].

## 4.2 Measuring Skin Erythema and Wheal Size

An injury, irritant or allergic reaction of the skin leads to reddening called erythema. There are several examples where a clinician will want to monitor this response, including recording a response to an irritating material or monitoring

spread of infection. Evaluation is generally done by a human, visually grading and tracing the response onto translucent paper [60] or measuring the largest diameter with a ruler [86]. Because of the irregularity and difficulties in identifying the boundaries of the wheal/erythema the reproducibility of a human observer judgments is limited [86]. The work in Chapter 2 and 3 was motivated by the example of detecting erythema and wheals after a skin-prick test. In this application additional information about the healthy skin and affected skin will be obvious to the clinician but the boundary may be ill-defined and difficult to outline. In this section we will discuss the application of the DSGS methods for improved detection of erythema and wheals as it relates to skin-prick and patch testing.

Skin-prick tests and patch tests are used by dermatologists and allergists to determine the patient's sensitivity (or allergic response) to an antigen or irritant. In skin prick testing, the patient is pricked with a liquid containing a small amount of an allergen and if a wheal with largest diameter greater than 3mm develops this is considered a positive reaction [64]. Figure 4.1 shows the results after a skin prick test on four participants reported in [64]. These participants have different skin phototypes as defined in Table 4.1. Some have suggested that the erythema that often develops around the wheal may be an important marker for improved prick test accuracy. Generally erythema is not measured or included as a metric in evaluating the test. [86] suggested that the absence of this data in testing is likely because detecting erythema in persons with darker skin is difficult.

A skin patch test can be used to determine skin sensitivity to an irritant. In this case the test material is taped onto the skin and when removed the clinician

visually grades and measures the response [62].

#### 4.2.1 Previous Work in Detecting and Measuring Erythema and Wheals

In practice both the skin prick tests and patch tests are evaluated by a human observer. There has been some work in computer-aided techniques to both measure the changes in erythema over time [60, 62, 91] as well as measure the wheal [64, 89] and erythema response [86]. These works relied on preprocessing the image. In the works of [64, 89], they both required that the wheals be outlined by a human and their only contribution was a better measurement of the area of the wheal versus the traditional measure of largest diameter.

Nischik and Forster [60] suggested that the best color space for measuring changes in erythema is the  $a$  axis of **CIELab**, calling it the “Erythema Axis.” Yamamoto et al. [91] used the EI index for improved measurement of erythema. A sequence of Adobe Photoshop manipulations including the “plastic wrap filter” and “poster edge filter” was suggested in [86] to enhance the wheal image in order to aid finding the area. In [91], in addition to detecting the erythema, they were interested in a quantitative measure of erythema and a metric for diagnostic purposes as well as a metric to measure changes over time. They introduced the metric  $\Delta EI$  which they defined as the mean over the unaffected patch minus the mean over the affected patch.



Figure 4.1: Image of skin after prick test on four different participants of different phototypes. Source: Rho Inc.

Table 4.1: **Fitzpatrick Phototype Scale:** This scale was developed in 1975 as a scale to measure the a patients skin phototype (sensitivity to UV radiation) and is the primary scale used today. This scale is often used by dermatologist as a scale of skin type/color [69].

Skin Phototypes	Skin Color	Reaction to Sun Exposure
Type I	white skin (very fair)	always burns, never tans
Type II	white skin (fair)	always burns, minimal tan
Type III	white skin (medium)	burns minimally, tans moderately and gradually
Type IV	light brown skin	burns minimally, tans well
Type V	brown skin	rarely burns, tans deeply
Type VI	dark brown/black skin	never burns, tans deeply



### 4.2.2 Motivation for Domain-Specific Grayscale Method

There are several distinct disadvantages to the past work presented for detecting erythema and wheals. The EI and  $a$  axis of **CIELab** were demonstrated as useful for Type II and Type III skin but cannot be adapted to other skin color types. Only the work of [86] was done on a variety of skin phototypes.

Lighting can have a significant effect on both the quality and the appearance of colors. This was partially accounted for in [91] where a reference white was included and used for white balancing of the images.

The processing in [86] was done in Adobe Photoshop where the filters used are intended for “artistic effect,” may change over time, and are provided without details.

The limitations mentioned above motivate the use of a new image-specific transformation of the color space. Clinicians may not be able to precisely determine boundaries but would be able to identify patches of affected and unaffected skin, making this real world problem an ideal application of our DSGS methods.

For this application (and the following) we only used the linear DSGS MAD method and the nonlinear LSVM DSGS method because we found after initial testing it was the most consistent nonlinear method for this application. The GSVM method was far too time consuming; for these applications it required many more support vectors than the LSVM method.

Table 4.2: Table of preprocessing methods for detecting and measuring erythema and wheals. See Section 4.2.4 and Section 4.2.5 for more details.

Method	Description	Advantages	Disadvantages
“Erythema Index” (EI) [91]	$EI = \log(\mathbf{R}) - \log(\mathbf{G})$	Based on model skin reddening. Heightens contrast of erythema in Type I-Type III skin.	Does not effectively detect wheal (not designed for). On Type V skin it failed to enhance erythema (Figure 4.11).
“Erythema Axis” [60]	$a$ axis of <b>CIELab</b>	Heightens contrast of erythema in Type I-Type III skin.	Does not effectively detect wheal (not designed for). On Type V skin it minimally enhance erythema (Figure 4.11).
Adobe Photoshop Filters [86]	“plastic wrap filter” and “poster edge filter” used to enhance erythema and detect wheal.	Method demonstrated as a diagnostic tool for determining positive allergic response over all skin types (not tested).	Details of filters are not published and may change over time. Filters were “hand-adjusted to accent the reaction” which can be time consuming and requires a knowledgeable eye.
DSGS MAD	Chapter 2, Section 2.3.1.2	In general outperforms the EI, “Erythema Axis” and grayscale image in contrasting the erythema and wheal.	Limited to linear relationships.
DSGS LSVM	SVM with $\log$ kernel (Chapter 3)	Significantly contrasted the area of interest over all methods tested.	Can be time consuming to compute and requires the choice of a kernel parameter.

### 4.2.3 Dataset

All the images used for this application are courtesy of Rho Inc. and were collected as part of the 2012 SAMSI Industrial Math/Stat Modeling Workshop for Graduate Students [64]. The images were taken after a skin-prick test was administered by an allergist. Images were captured on three different cameras including a CanonS95 (image number starting with a 2), CanonSX210(image number starting with a 1) and a iPhone 4 (image number starting with P). From this dataset we created two subsets, one for the detection of erythema (Figure 4.2) and one for measuring the wheals (Figure 4.3). All images were cropped to highlight the areas of interest. We had a human user hand-draw the boundary around the wheal of W4 and W5 as seen in Figure 4.4. The boundaries were drawn by Anna Gleason in ADOBE ILLUSTRATOR based on the color image.

For our training set we hand selected the background and foreground regions

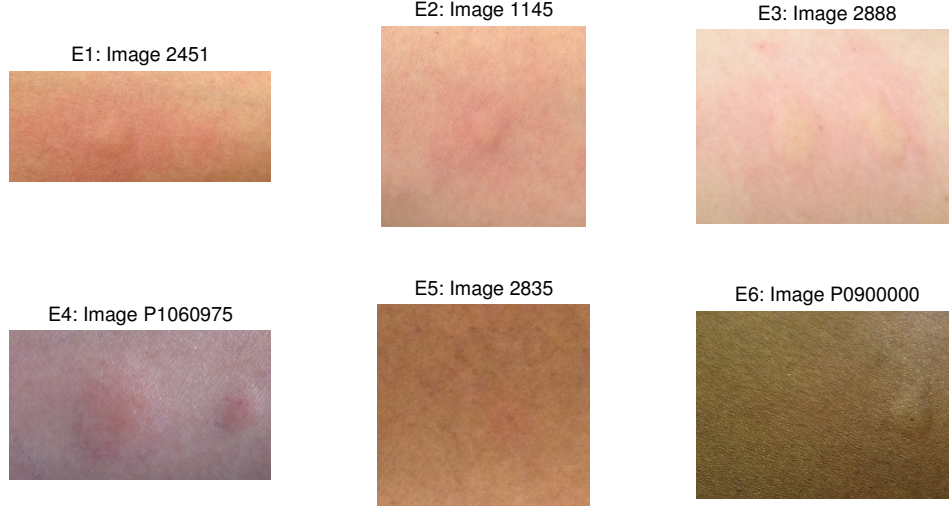


Figure 4.2: Six images used for measuring erythema after prick test. Source: Rho Inc.

that capture the areas of interest and reflect a selection a practitioner could make, for example, on a mobile device.

#### 4.2.4 Summary of Results of Measuring Erythema

In Figures 4.6 - 4.11 we show the resulting DSGS images compared to the previous suggested single-coordinate images for detecting erythema. In Table 4.2.4 we show the relative change in erythema (4.6) which we measure as

$$\tilde{\Delta}\text{EI} = \frac{|\mu_{\mathbf{P}} - \mu_{\bar{\mathbf{P}}}|}{\max(\mathbf{X}) - \min(\mathbf{X})}, \quad (4.8)$$

where  $\mathbf{P}$  and  $\bar{\mathbf{P}}$  are the foreground and background patches respectively and  $\mathbf{X}$  is the entire image. The  $\tilde{\Delta}\text{EI}$  helps indicate how well the area of interest is contrasted. From the results we see that all the presented methods improve upon the traditional Grayscale. In the improved DSGS images the erythema is more visually

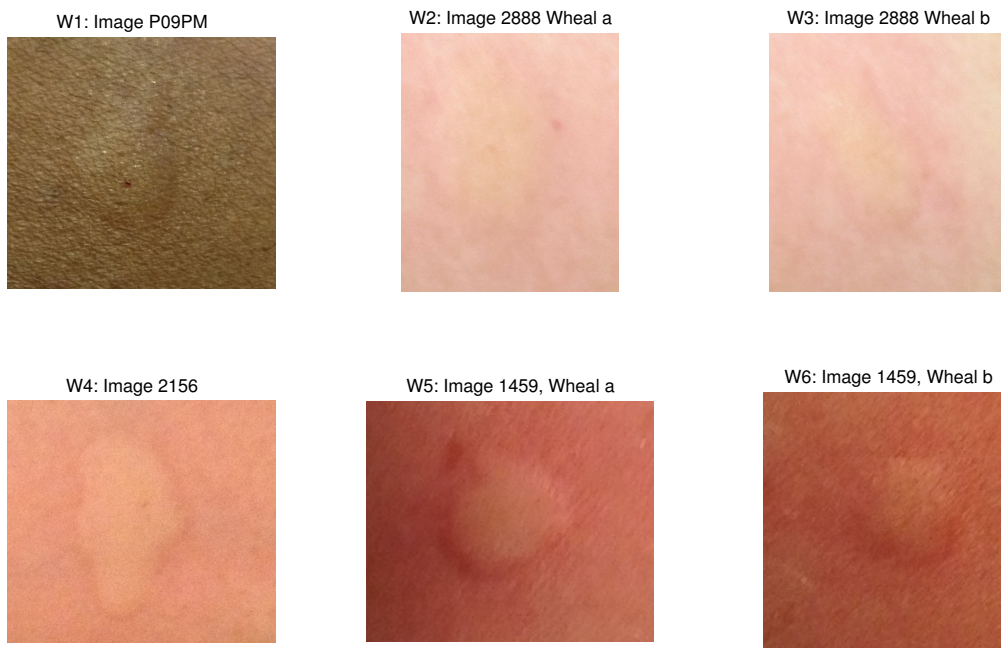


Figure 4.3: Six images used for measuring wheal after prick test. Source: Rho Inc.

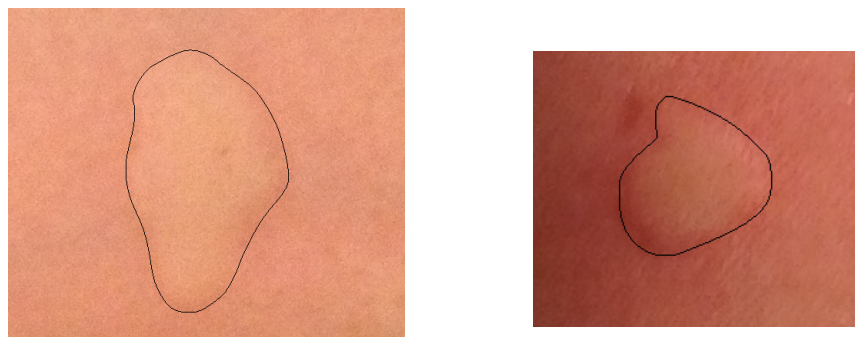


Figure 4.4: W4(left) and W5 (right) with hand-drawn boundary of wheal. Source: Rho Inc. where the boundaries were drawn by Anna Gleason.

apparent.

Despite having additional information our linear MAD DSGS method did not provide much improvement over the unaided methods when compared by relative  $\Delta\tilde{EI}$ . This suggests the relationship between the background and foreground is not represented well by a linear model. In Figure 4.12 we show the scatter plots of the background and foreground pixels and we can clearly see from these that for the images E2 - E4 we have a distinct nonlinear relationship. The nonlinear relationship is further confirmed when we compare the linear MAD DSGS to the nonlinear LSVM DSGS method. We see that the nonlinear DSGS method shows significantly more contrast between erythema and the unaffected skin when compared to the other methods presented.

If we compare the edges found after applying the Chan-Vese level set method on the single-coordinate images found for E2, E4, and E5, as seen in Figure 4.13, we see that both the linear and nonlinear DSGS methods found very similar edges compared to the “a” of Lab, LogG-LogR (not shown), and the traditional grayscale image.

These results suggest that the DSGS methods could be a powerful tool used to aide human detection of erythema as well as for inclusion in an automated system for detecting and measuring the area of erythema.

The nonlinear LSVM DSGS method produced visually more impressive images but there are significant challenges when using nonlinear DSGS method in real-world scenarios. First, for the images in this dataset, a successful nonlinear DSGS method is dependent on the choice of background/foreground that captured all

variations in the skin and illumination. In Figure 4.5 we show the effect of two different choices of seemingly similar user supplied backgrounds. In this case the smaller selected background does not capture all the changes in slight shading caused by the lighting and contour of the forearm. The nonlinear DSGS method fails to correctly identify the darker regions of the unaffected skin (note that this image is displayed so that all the negative values are displayed as black in order to highlight the pixels in the foreground). The effects of the different background selection had much less impact on the linear DSGS method. This can be accounted for because pixels that are closely related (as measured by Euclidean distance) will stay closely related in all orientations. Second, for each of the images in this training set, LSVM took significantly more time to transform the image compared to each of the other methods. The time for the DSGS methods are given in Table 4.2.4 (all other presented methods took no more than a second to compute). The time that it takes to compute the LSVM was dependent on the number of support vectors which is much more dependent on the characteristics of the training data than the size of the training data or color image. Table 4.2 includes a list of each method tested or discussed and a brief discussion of advantages and disadvantages.

#### 4.2.5 Summary of Results of Measuring Wheal

In Figures 4.14 - 4.19 we show the resulting images of the six test images for detecting the wheals. Both the linear and nonlinear DSGS produce results with significantly better contrast between the raised area and the unaffected skin

Table 4.3: Relative  $\Delta EI$  for each method of the erythema images.

Image	GS	a of CIELAB	log R - log B	MAD	LSVM $d = 1$
E1	0.0697	0.3290	0.2181	0.3904	0.6546
E2	0.1040	0.3010	0.2289	0.3055	0.5612
E3	0.1060	0.4010	0.3430	0.3991	0.7564
E4	0.1162	0.3254	0.3290	0.2875	0.6483
E5	0.0379	0.5189	0.3075	0.4815	0.6762
E6	0.0106	0.1194	0.0326	0.1512	0.4407

Table 4.4: Time to compute DSGS for the erythema images in seconds.

Method	E1	E2	E3	E4	E5	E6
MAD	3.8	2.1	2.9	3.1	1.6	4.8
LSVM	2269.9	62.0	5.5	412.8	2.7	-

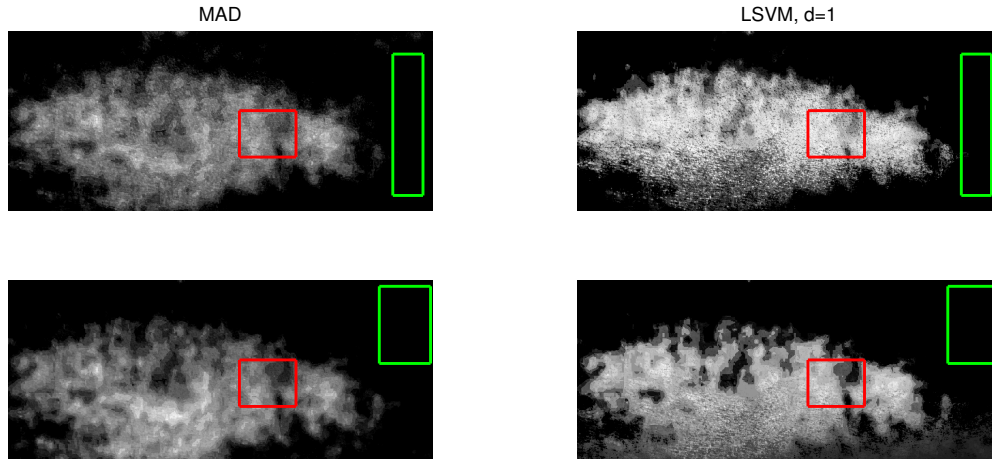


Figure 4.5: E1: Comparing the choice of training background. Displayed such that pixels less than zero are displayed as black to emphasis on the classification of pixels.

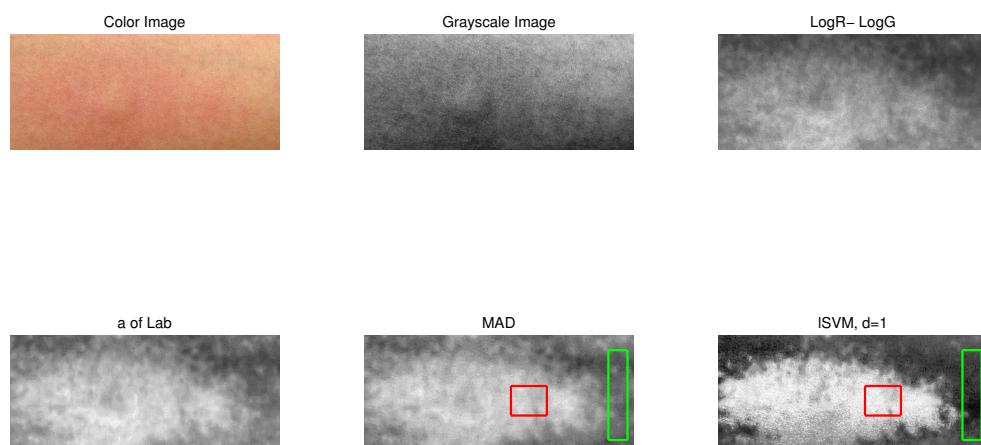


Figure 4.6: E1: GS, Log,a of Lab, MAD, and LSVM.

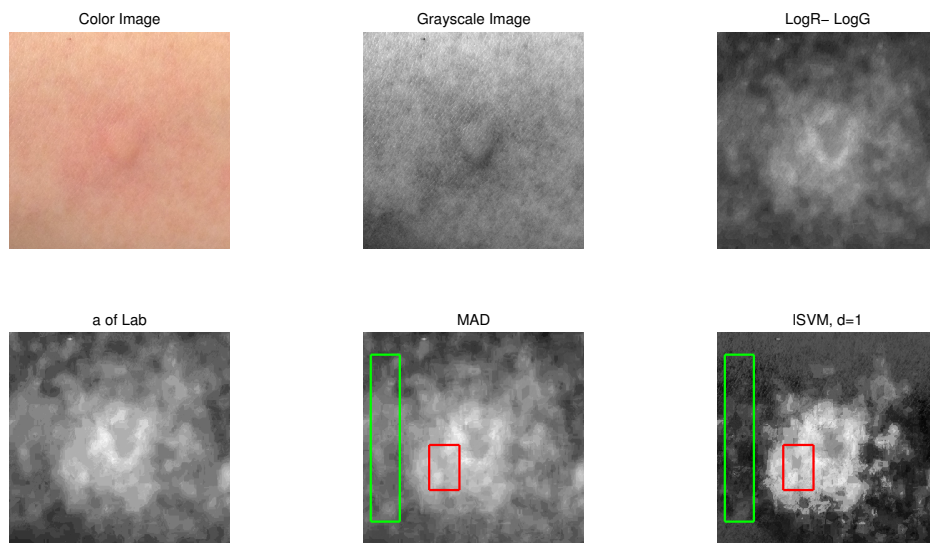


Figure 4.7: E2: GS, Log,a of Lab, MAD, and LSVM.



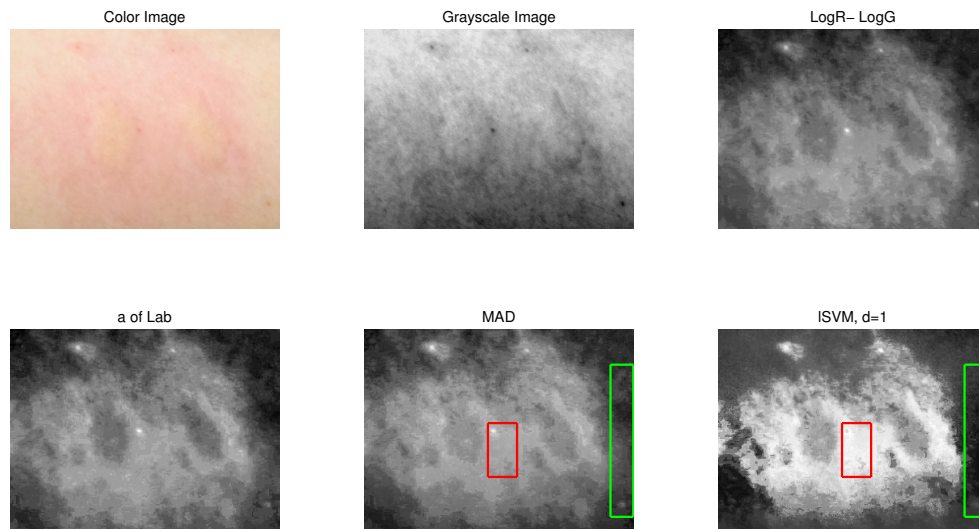


Figure 4.8: E3: GS, Log,a of Lab, MAD, and LSVM.

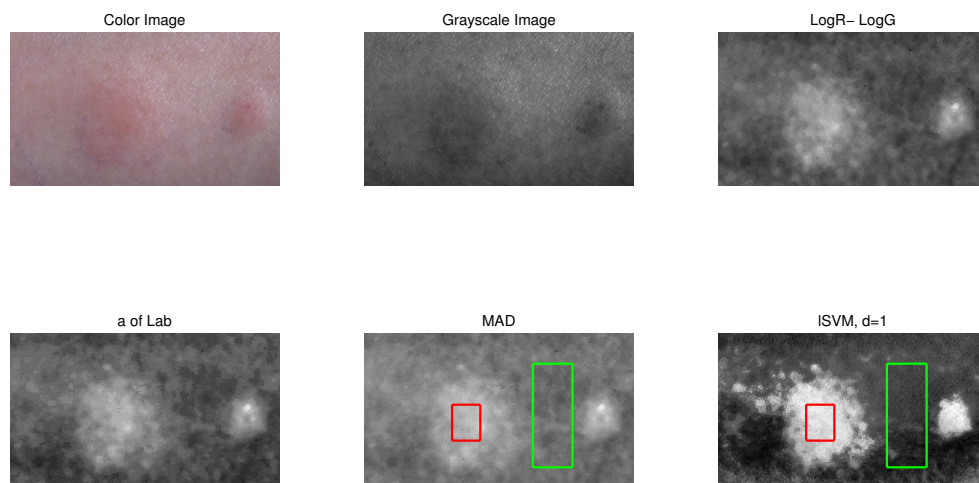


Figure 4.9: E4: GS, Log,a of Lab, MAD, and LSVM.

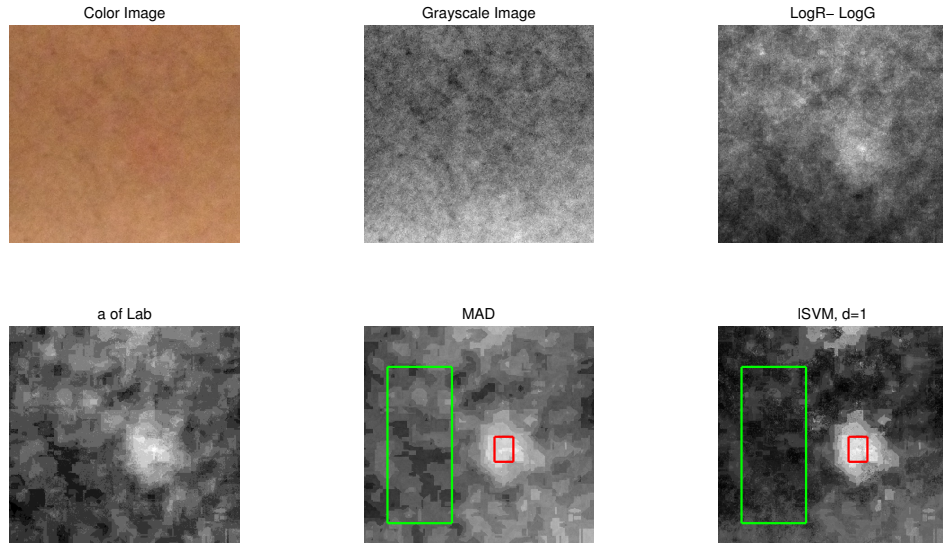


Figure 4.10: E5: GS, Log,a of Lab, MAD, and LSVM.

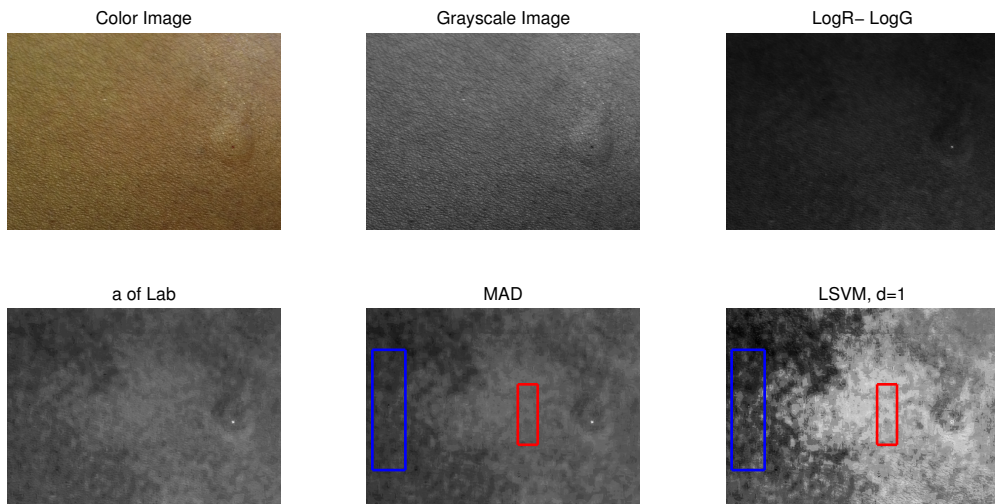


Figure 4.11: E6: GS, Log,a of Lab, MAD, and LSVM.

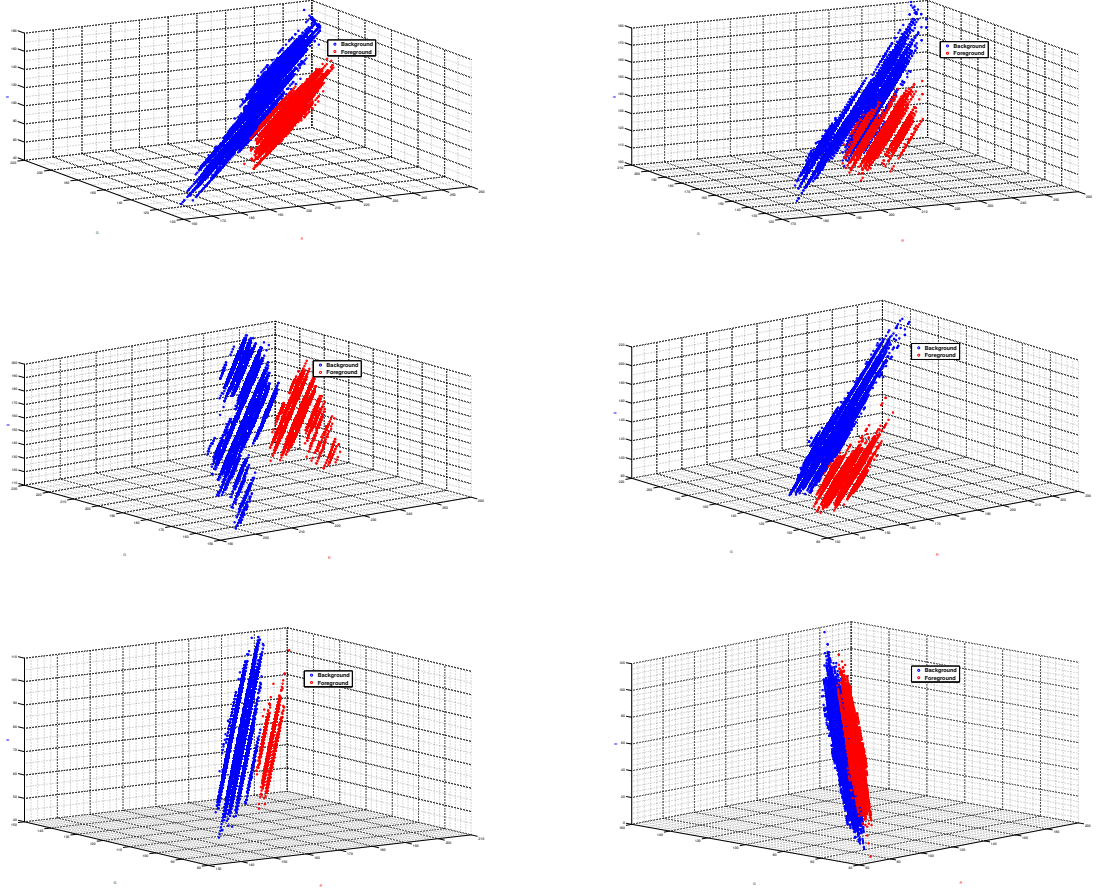


Figure 4.12: Scatter plot of pixels of  $P$  (red) and  $\bar{P}$  (blue) in **RGB** space for images E1-E5.

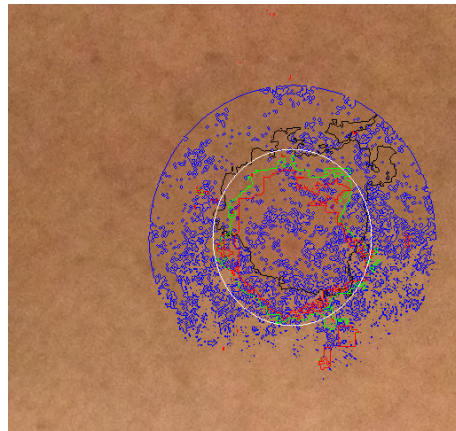
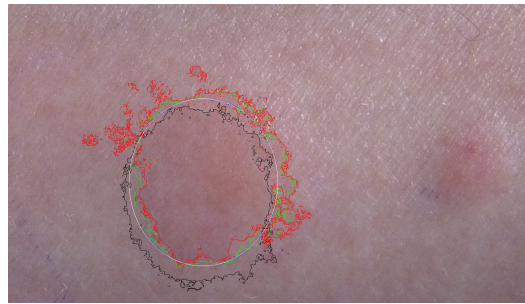
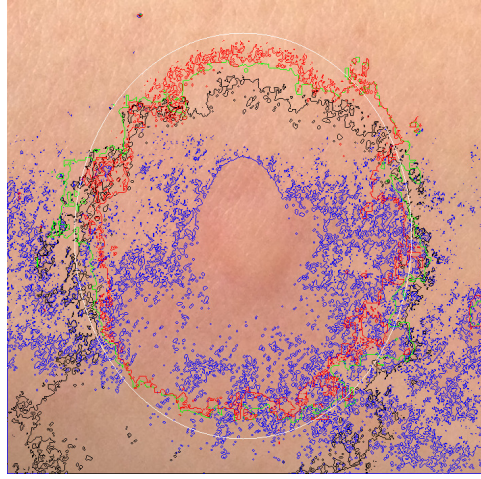


Figure 4.13: E2, E4, and E5. Resulting boundary after Chan-Vese to Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), “a” of Lab (black), and GS (blue).

compared to the other methods.<sup>1</sup> Again the nonlinear DSGS method produces an image with far more contrast suggesting a nonlinear relationship between the wheal and the un-raised patch. In general we found the methods suggested for detecting erythema are not suitable for detecting the wheals. In Figure 4.20 we see the scatter plot of the selected background and foreground pixels and we see a range of relationships.

Visually the grayscale images appear to best capture the boundaries of the raised area. Recall that traditional grayscale is a model of light intensity. In this case we are detecting a raised area of skin where there are shadows around the boundary. The grayscale image captures those shadows but loses information about the subtle changes in color. On the other hand the DSGS methods capture changes in color but seem to lose information about shading. This is particularly apparent in the results of ‘Wheal 5’ as seen in Figure 4.18.

The boundaries found after applying the Chan-Vese are seen superimposed on the color images in Figure 4.21. The grayscale images failed to produce any relevant edges (in general the edges found were influenced by changes in illumination). Similar to the application for detecting the erythema, both the linear and nonlinear DSGS produce very similar boundaries. That being said the boundaries do not match what the human eye would determine as the boundary as demonstrated in Figure 4.22. We see that the boundary based on the DSGS image captures the

---

<sup>1</sup>Although measuring the wheal is a different task than measuring erythema, we included the two suggested erythema single-coordinate methods ( $a$  axis of **Lab** and  $\log R - \log G$ ) since the wheals are generally surrounded by erythema.

Table 4.5: Time to compute DSGS methods for the wheal images in seconds.

<b>Method</b>	W1	W2	W3	W4	W5	W6
MAD	1.9	0.6	0.7	1.3	1.0	1.6
LSVM	1046.3	6.3	749.2	2503.1	9.1	64.1

general shape of the wheal but do not match the human-drawn boundaries and do not capture the transition from the un-raised skin to the raised skin (wheal). This may or may not be significant. If the area found were off by a constant factor this discrepancy may not be a problem but would require trials to determine whether the measured wheal size indicates a positive test. The ability of the DSGS images to find better boundaries could be improved on if the images were captured in a way that reduced shadows. In the resulting image, the difference between the wheal and un-raised skin could then be determined by the subtle color changes.

In Table 4.2.5 we again see that the time to compute the nonlinear LSVM DSGS method is significant suggesting that the linear DSGS may be computationally robust when incorporated into an application or device.

Further work needs to be done to test the viability of using the DSGS methods for measuring the area of wheals after skin-prick tests. These results show some promise for the future of an automatic wheal measuring device which was the initial goal of the work by [64]. Table 4.2 includes a list of each method tested or discussed and a brief discussion of advantages and disadvantages.



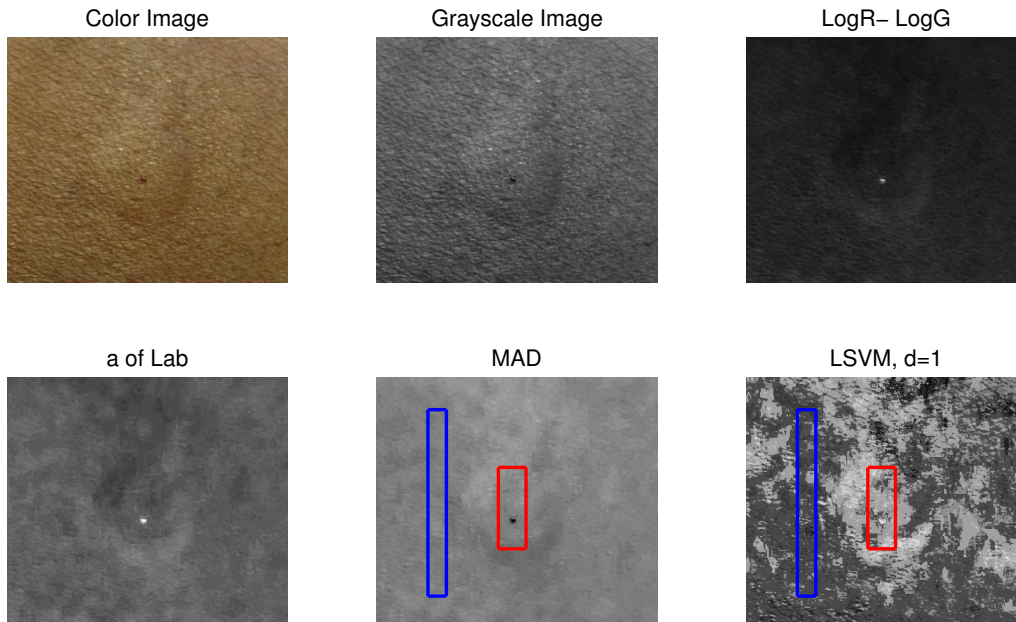


Figure 4.14: W1: GS, Log,a of Lab, MAD, and LSVM.

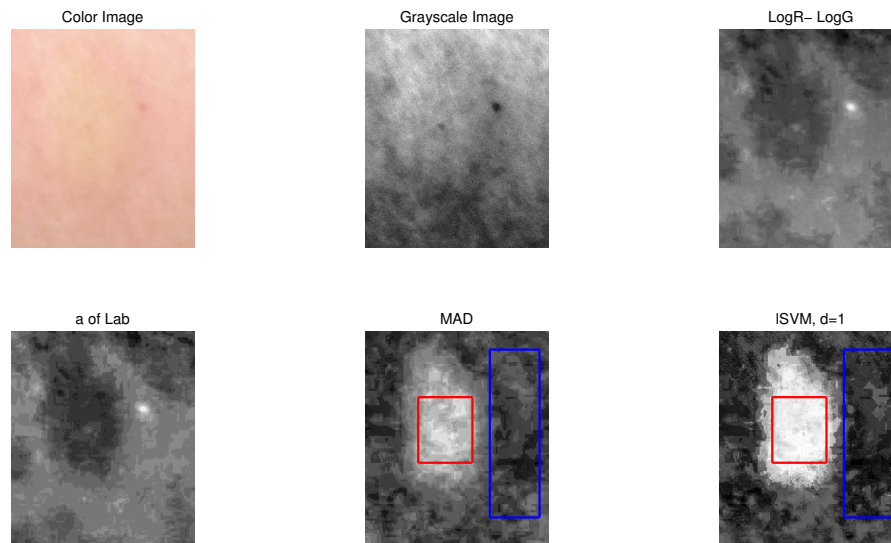


Figure 4.15: W2: GS, Log,a of Lab, MAD, and LSVM.

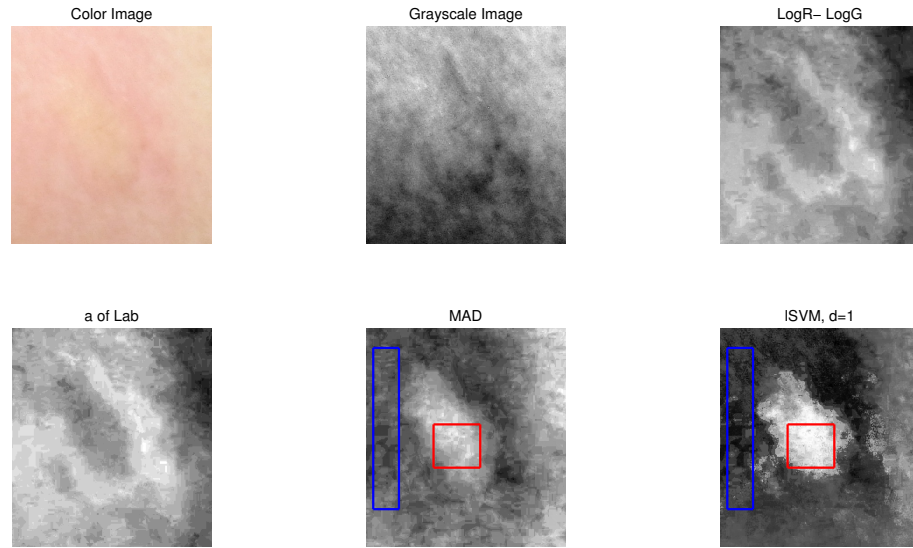


Figure 4.16: W3: GS, Log,a of Lab, MAD, and LSVM.

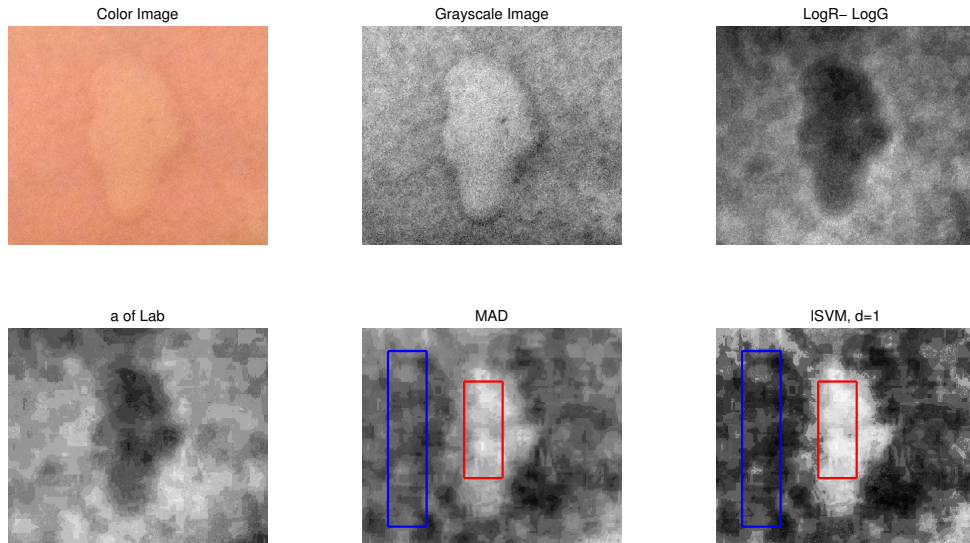


Figure 4.17: W4: GS, Log,a of Lab, MAD, and LSVM.



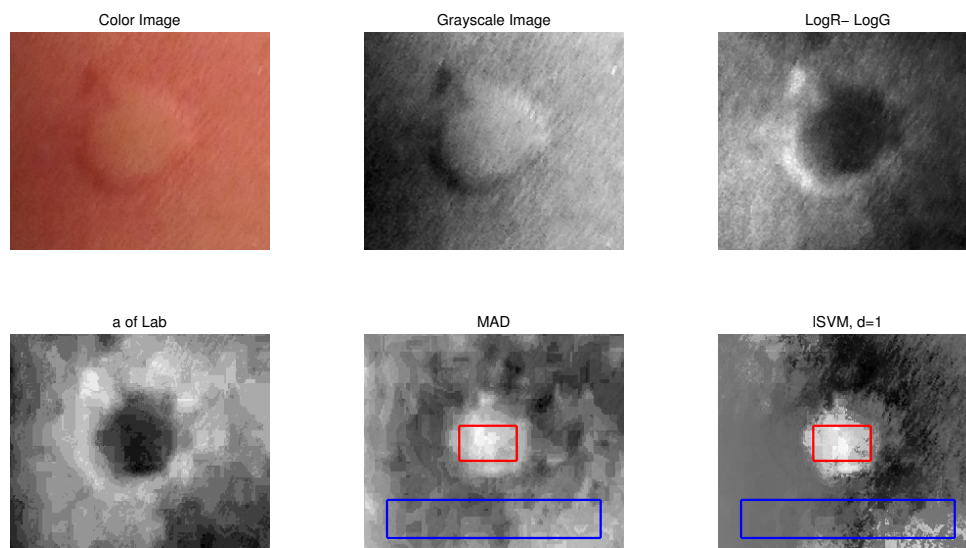


Figure 4.18: W5: GS, Log,a of Lab, MAD, and LSVM.

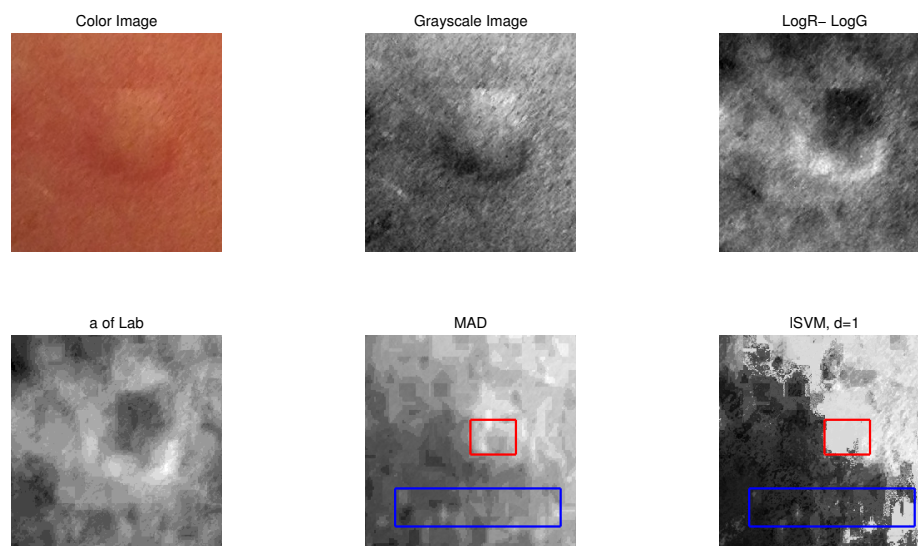


Figure 4.19: W6: GS, Log,a of Lab, MAD, and LSVM.

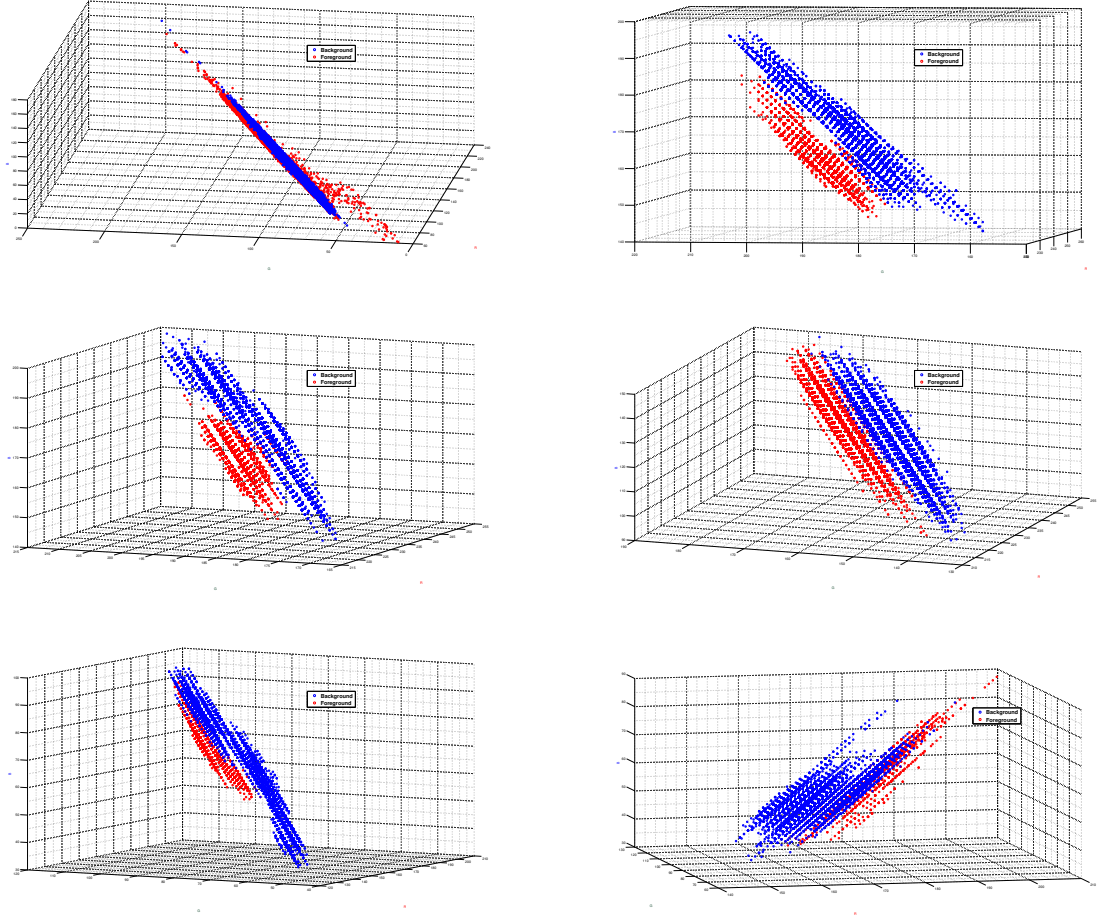


Figure 4.20: Scatter plot of pixels of  $P$  (red) and  $\bar{P}$  (blue) in **RGB** space for images W1-W6.

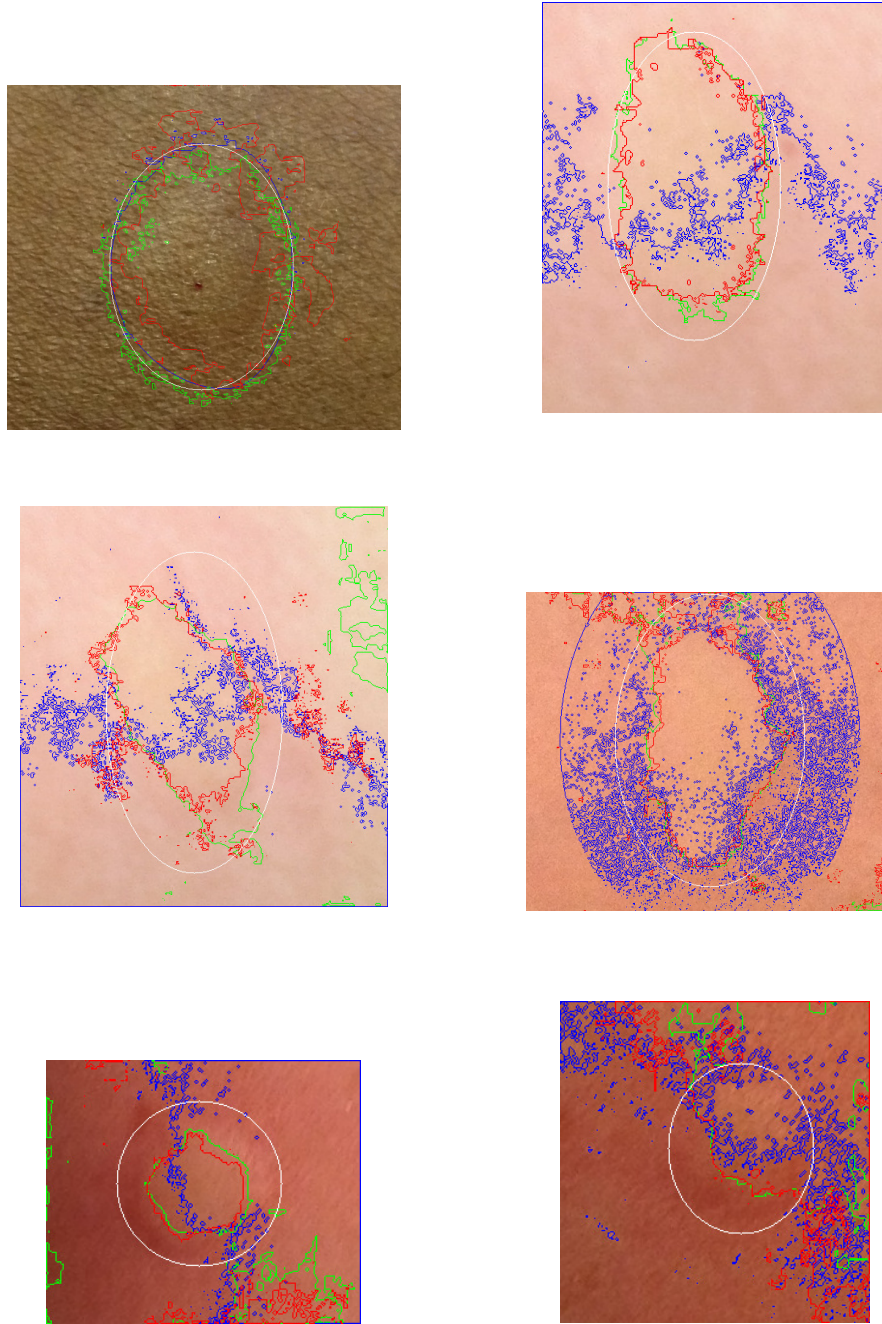


Figure 4.21: Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), and GS (blue) for W1- W6.



Figure 4.22: Image comparing the human drawn edge to the resulting boundary of the DSGS image for W4 (left) and W5 (right). Hand-drawn (black), LSVM (red), and MAD (green).

### 4.3 Border Detection of Skin Lesions

Skin cancer is the most common type of cancer in the United States with over 3.5 million cases diagnosed in 2013. The most deadly form of skin cancer is invasive melanoma. Invasive melanoma only accounts for 5% of all skin cancers, but it accounts for the majority of skin cancer related deaths (12,650 deaths in 2013). Early intervention is key to invasive melanoma survival, where the survival rate is 98% for cases diagnosed before spreading to the lymph nodes or other organs [1]. Unfortunately, detecting melanoma (as well as other skin cancers) in its early stages is often difficult because differentiating melanoma from benign lesions is error prone, even for an experienced dermatologist. Figure 4.23 shows some of the variety of colors and shapes of skin lesions both benign and cancerous.

Dermatologists look for changes in a skin lesion, checking any new skin lesions with the ABCD criteria [68], which stands for

- (A)symmetry growth,
- Irregularity in the (B)order (jagged or notched),
- Nonuniform (C)olor,



Figure 4.23: Skin Lesions, variety of colors. First two are moles, second two are cancerous. Source: First two images are from the Dermoscopy Atlas. Second two images are from DERMOFIT.

- (D)iameter greater than 6 millimeters.

In addition to inspection by a trained dermatologist's naked eye, common practice in dermatology includes the use of dermatoscope and the taking of baseline images to aid in early detection of skin cancer. Computer-aided diagnostic instruments are becoming increasingly available to dermatologists. These new technologies often compare a captured image to a large training dataset. In many of these algorithms, the first step is finding the boundary of the lesion [78]. A good boundary accurately locates the lesion and reduces the irrelevant information for improved classification. Characteristics of the lesion such as area, symmetry, and irregularity of the boundary of the lesion can also be gained from an accurate boundary and can improve diagnosis of cancerous lesions. A number of the algorithms have quantifications of the ABCD criteria or competing criteria. In this work we will show how the DSGS methods can be applied, greatly improving the process of finding the boundary and segmenting the lesion of interest.

### 4.3.1 Past Work in Skin Lesion Boundary Detection

There is existing research in the area of boundary detections and segmentation of images of lesions on the skin. The first step in finding the boundary is generally preprocessing the image and finding a single-coordinate image that enhances the lesions [51]. A number of different transformations of color images have been suggested for the application of detecting and segmentation skin lesions. These include using the **B** channel of **RGB** image, grayscale, the first coordinate after applying principal component analysis (PCA) (coordinate with most variation), the **b** axis of **CIELab**, and the approximated Melanin Index of multispectral images [51].

The work by [30] suggested using the **B** channel of **RGB**, the **b** axis of **CIELab** and 3D clusters in XYZ space in combination for improved segmentation of melanoma images. In [32] they suggested a technique with some similarities to the new MAD method presented in this work; they assumed the data was bimodal and found the orientation that maximized the concavity in the bimodal mixture model. In the work by [21] they used the melanin index of multispectral images where the melanin index is given by

$$A_1 - A_2 = \log(1/R_1) - \log(1/R_2) \quad (4.9)$$

for wavelength  $\lambda_1 = 650nm$  and  $\lambda_2 = 700nm$ . In the absence of multispectral images we will compare our improved methods to the approximate melanin channel as suggested by [91] to be

$$MI = -\log(\mathbf{R}). \quad (4.10)$$

Table 4.6: Table of preprocessing methods for border detection of skin lesions. See Section 4.3.1 for more details.

Method	Description	Advantages	Disadvantages
“Melanin Index” (MI) [91]	$MI = -\log(R)$	Based on the skin layer model. In [21] they used multispectral data to find the MI and showed it performed well on a limited number of examples.	Often was outperformed by grayscale suggesting that the approximate MI index as suggested by [91] is not a good approximation of the Melanin.
PCA (also called Karhunen-Loeve) [26]	Principal axis of PCA		Very sensitive to skin and lighting variations.
<b>B</b> channel [51]	<b>B</b> channel of <b>RGB</b>	Seen to enhance lesions of brown tints and light skin.	Failed to enhance lesions of pink and blue tones. Can not adapt to skin tones, speculate that this method would have a hard time enhancing lesions on Type IV-VI.
<i>b</i> of CIELAB [30]	<i>b</i> axis of <b>CIELab</b>	Detects the brown and blue color lesions on light skin.	Failed to enhance lesions of pink and red tones on light skin. Can not adapt to skin tones, speculate that this method would have a hard time enhancing lesions on Type IV-VI.
DSGS MAD	Chapter 2, Section 2.3.1.2	In general outperforms the methods above.	Limited to linear relationships
DSGS LSVM	SVM with <i>log</i> kernel (Chapter 3)	Significantly contrasted the area of interest over all other methods.	Can be time consuming to compute and requires the choice of a kernel parameter.

### 4.3.2 Motivation for Domain-Specific Grayscale Method

Similar to the application of detecting skin erythema and wheals the user can easily supply information about the background and foreground suggesting that this is a good application of the DSGS methods. In addition to challenges of variations in skin color and lighting conditions in this application there can be a lot of variation in the color of the lesions themselves suggesting a domain specific solution.

### 4.3.3 Dataset

For this work we used two types of images, images taken by a commercial camera with a standard lens and images taken using a dermatoscope. All the dermatoscope images, seen in Figure 4.24, were sourced from the Dermoscopy Atlas

with permission from the editor [20]. The images taken with a standard lens, Figure 4.25, came from the Dermoscopy Atlas and DERMOFIT. The DERMOFIT research project has collected a dataset of images of lesions that they provide for public use [27].

#### 4.3.4 Summary of Results on Border Detection of Skin lesions

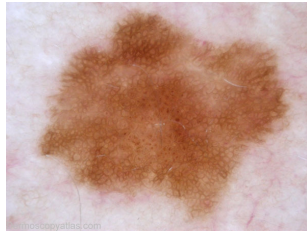
In this application we were interested in finding the boundary of skin lesions as this is generally the first step to identifying the type of lesion. In this case we compared our DSGS methods to the previous cited single-coordinate images for skin lesion identification. Unlike in the previous application for identifying wheals and erythema, where each of the images are very similar and have boundaries with similar characteristics, these lesions are significantly different from each other and have boundaries with a range of characteristics.

In the case of the images captured with a dermatoscope, the nonlinear LSVM DSGS method produced results with significant contrast between the lesions and background skin compared to the linear DSGS method and the other presented methods as seen in Figures 4.26 - 4.32. The linear MAD DSGS method produced much more consistent results compared to most of the methods suggested in the literature. In this dataset  $b$  of **Lab** also produced consistent results but given a larger dataset with a variety of lesions and skin phototypes we suspect there will be cases where it will not find contrast between the lesion and surrounding skin.

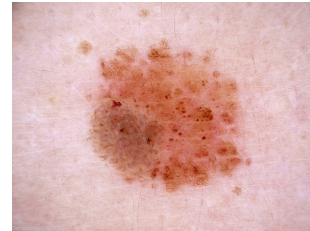
In Figure 4.33 we show the results of finding the boundary using the Chan-



D1: Clark Nevus



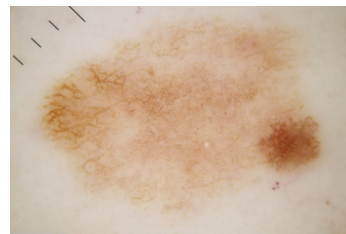
D2: Clark Nevus with Seborrhoeic Keratosis



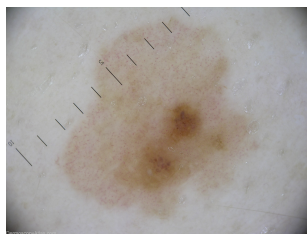
D3: Acral Melanocytic



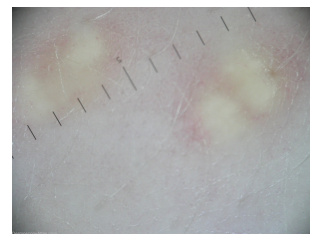
D4: Blue Nevus



D5: Nevus Dysplastic



D6: Melanoma Amelanotic



D7: Eruptive Xanthroma

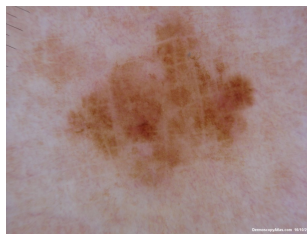


Figure 4.24: Dermatoscope images of a variety of skin lesions. Source: Dermoscopy Atlas.

S1: Acral Melanocytic



S2: Basal Cell Carcinoma



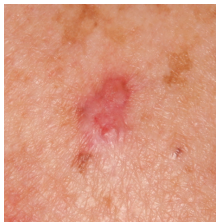
S3: Dermatofibroma



S4: Intraepithelial Carcinoma



S5: Actinic Keratosis



S6: Melanocytic



S7: Malignant Melanoma



S8: Seborrheic Keratosis



Figure 4.25: Standard photo taken on a digital camera. Source: Top left from Dermoscopy Atlas. Remaining from DERMOFIT.

Table 4.7: Time to compute DSGS methods for the Dermatoscope images.

Method	D1	D2	D3	D4	D5	D6	D7
MAD	0.9	3.3	6.5	3.1	3.2	3.8	2.7
LSVM	0.6	5.0	244.8	3.8	4.2	5.8	2.7

Table 4.8: Time to compute DSGS methods for the standard lesion images.

Method	S1	S2	S3	S4	S5	S6	S7	S8
MAD	1.0	0.6	1.4	2.3	0.8	0.8	0.9	0.6
LSVM	0.3	0.8	144.8	4.1	2.1	1.3	1.8	0.9

Vese method. For D1-D4 the nonlinear and linear DSGS methods produce very similar boundaries. For D5-D7 only the nonlinear DSGS method found boundaries that matched the perceived boundaries. In this case, because of the high contrast in the nonlinear DSGS images, simply finding the segmentation of the image based on the SVM values would be sufficient to find the boundary of the lesion.

For the images taken without a dermatoscope, the linear DSGS method performed close to the level of the nonlinear DSGS method, as measured by the visual contrast (Figures 4.34 - 4.41) and the edges found after applying the Chan-Vese method (Figure 4.42). This is somewhat surprising as these images are much more likely to be affected by differences in lighting and setting. This could be because these images do not contain the internal patterns/structures of the lesions that the dermatoscope magnification provides and they appear more uniform benefiting the linear method. Table 4.6 includes a list of each method tested and a brief discussion of advantages and disadvantages.

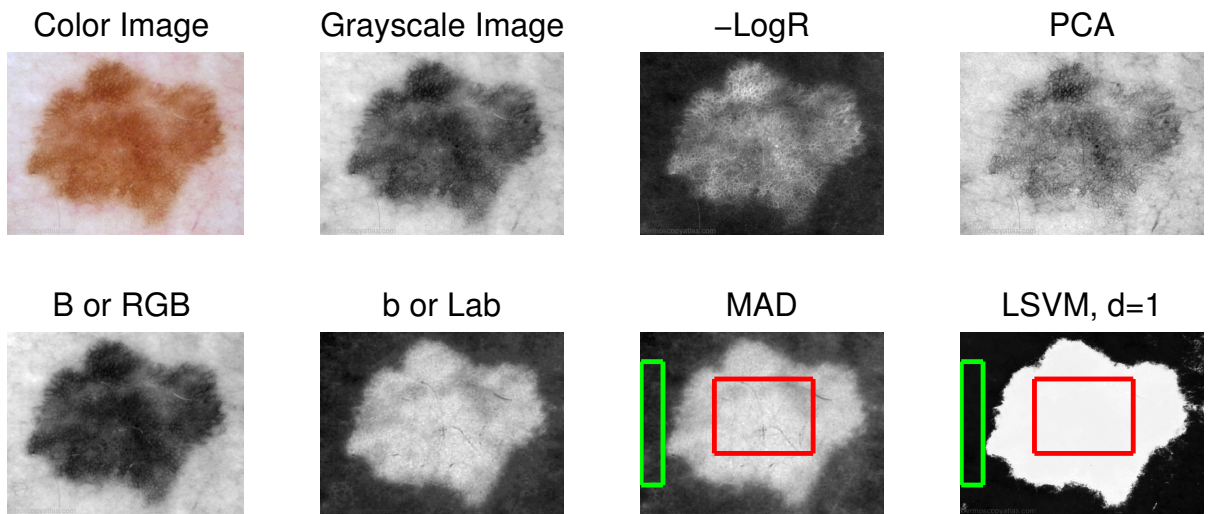


Figure 4.26: D1- Top: Color, GS, -LogR. Bottom: “B” or RGB, “b” of Lab, MAD, and LSVM.

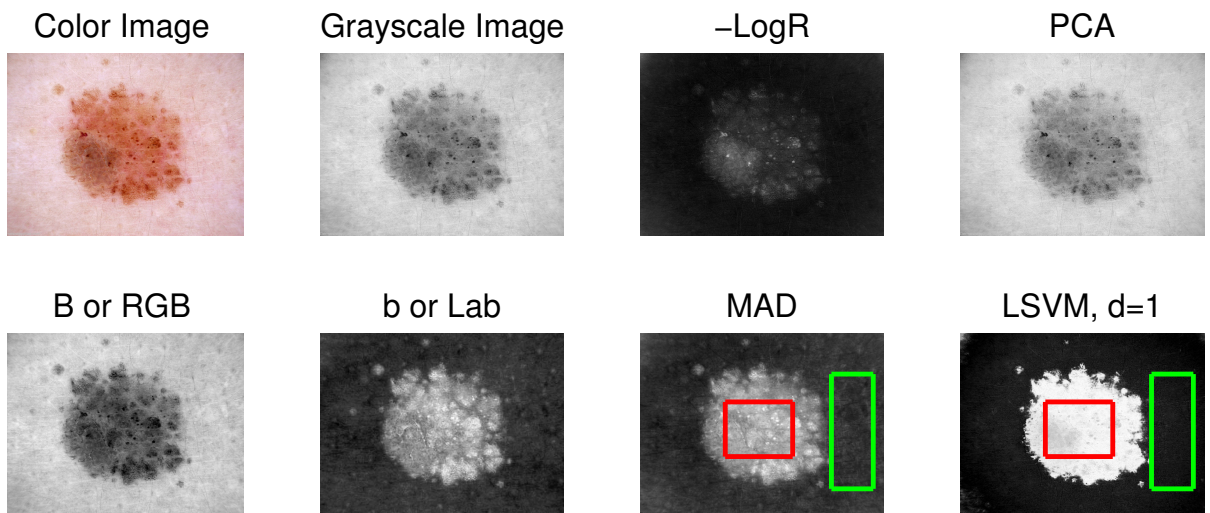


Figure 4.27: D2- Top: Color, GS, -LogR. Bottom: “B” of RGB RGB, “b” of Lab, MAD, and LSVM.

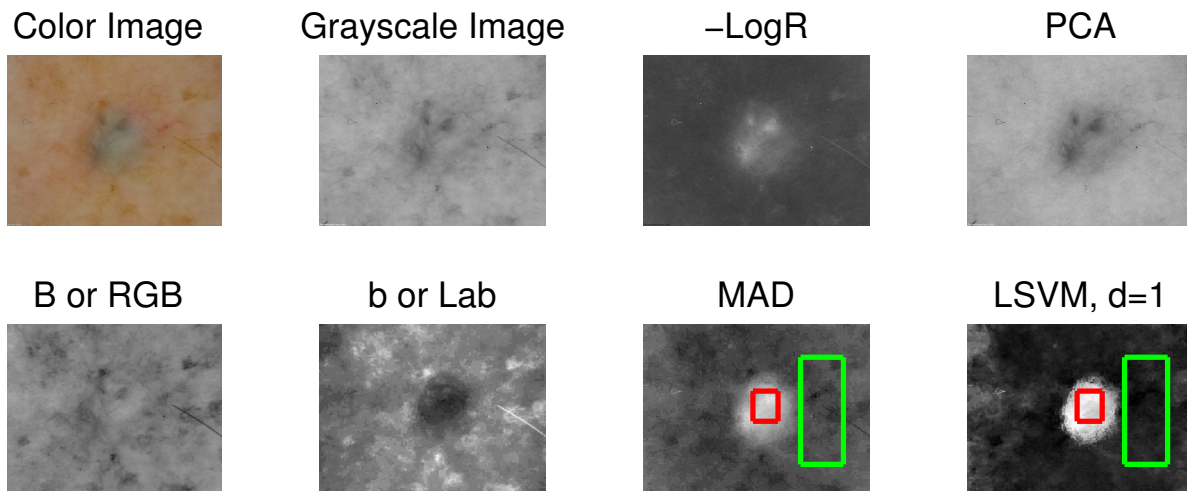


Figure 4.28: D3- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

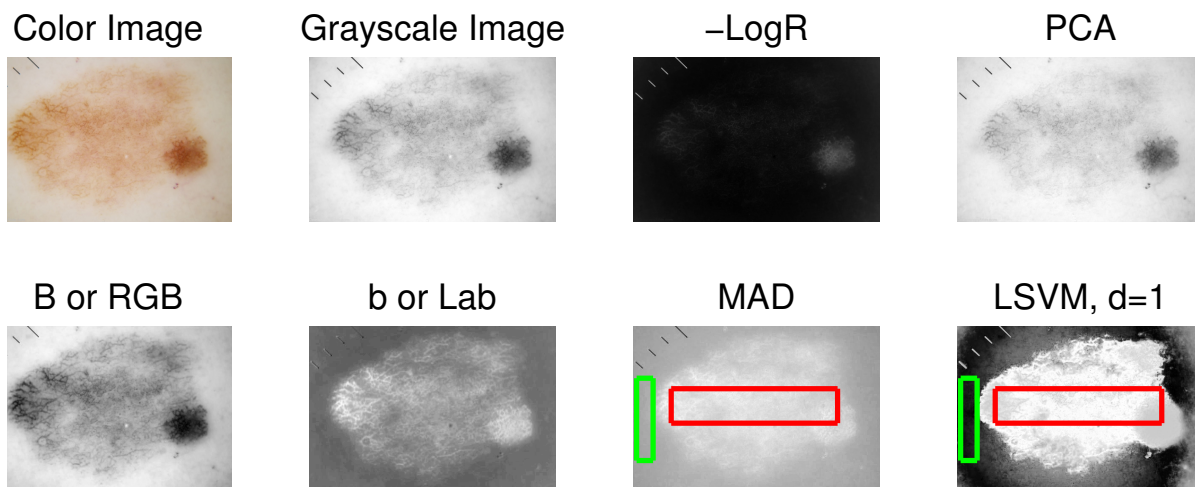


Figure 4.29: D4- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

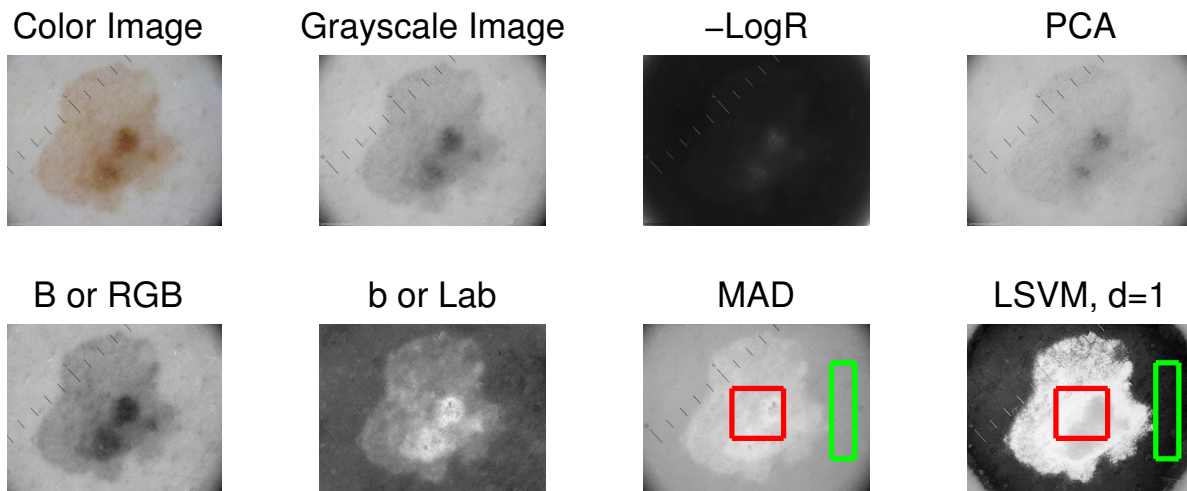


Figure 4.30: D5- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

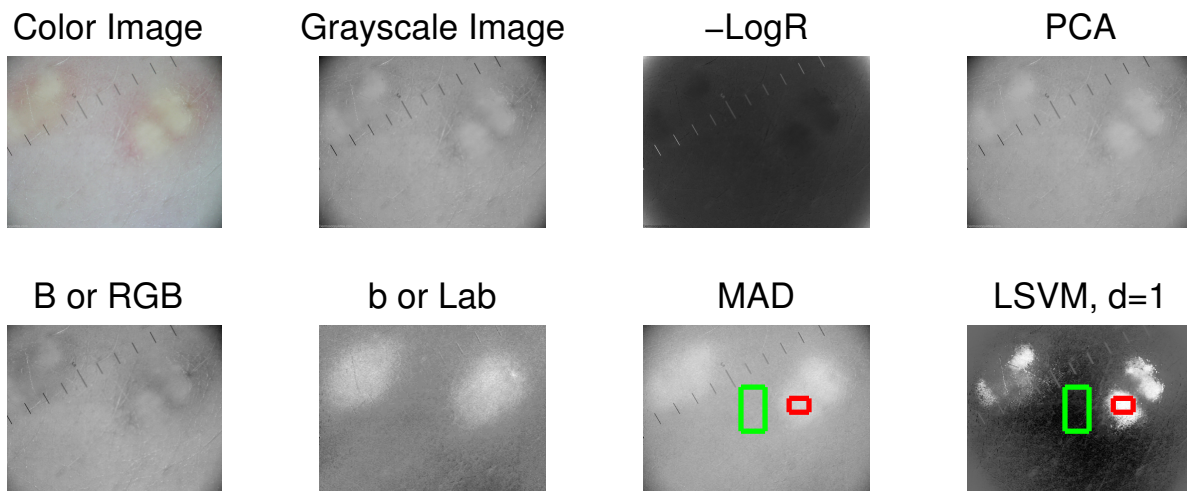


Figure 4.31: D6- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.



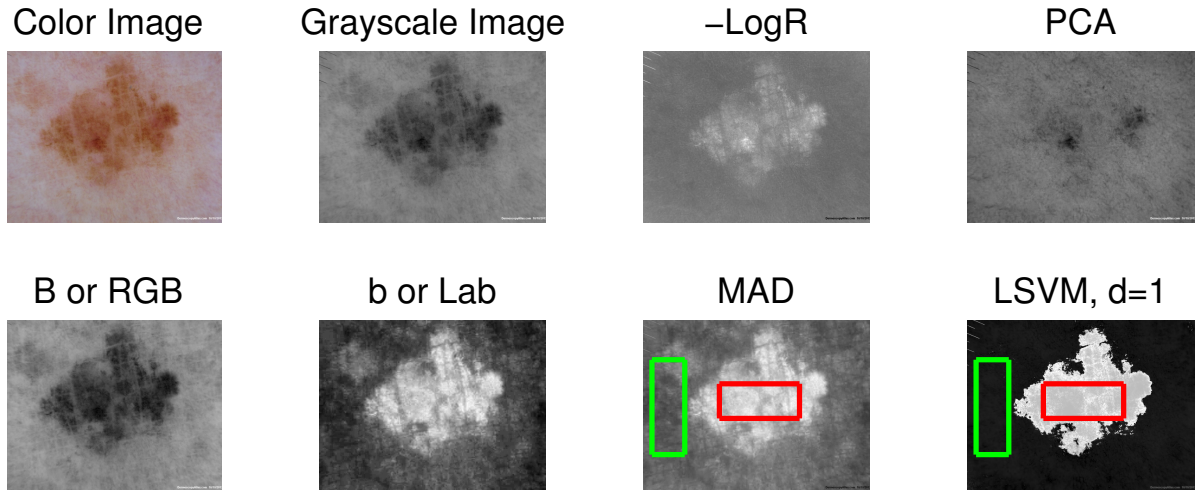


Figure 4.32: D7- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

#### 4.4 Summary and Conclusion

We saw in this chapter that the DSGS methods have the potential for significantly improving results for some of the tasks that arise in dermatology related to digital image use. In this application, when compared visually, the nonlinear methods produce images where the object of interest was much more enhanced suggesting that the nonlinear DSGS method can improve the detection of erythema, measuring wheals after skin-prick tests, as well as detecting and finding the boundary of skin lesions. In the case of measuring the wheal, we found that the boundary found using the linear and nonlinear DSGS images were consistent in the general shape as compared to the hand-drawn images.

Despite visually impressive images found by using the nonlinear LSVM DSGS method, the method is much more dependent on user supplied information than the

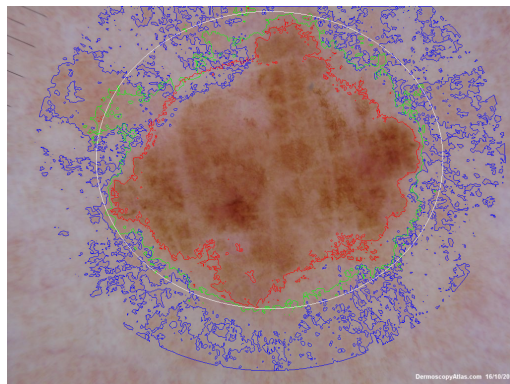
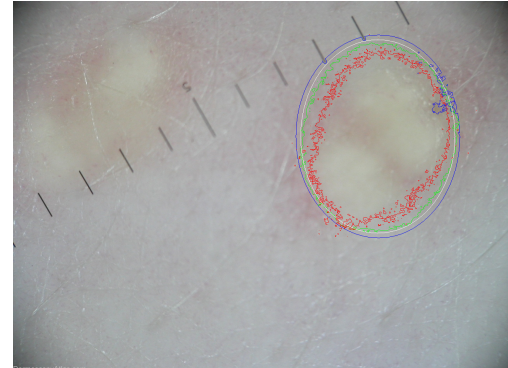
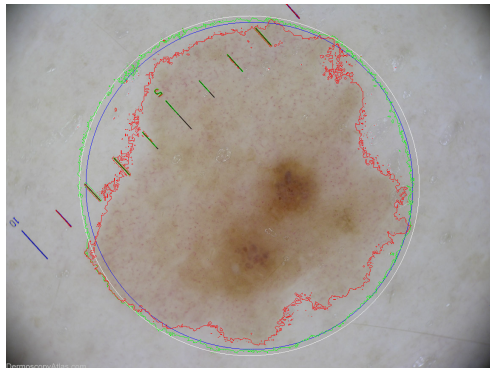
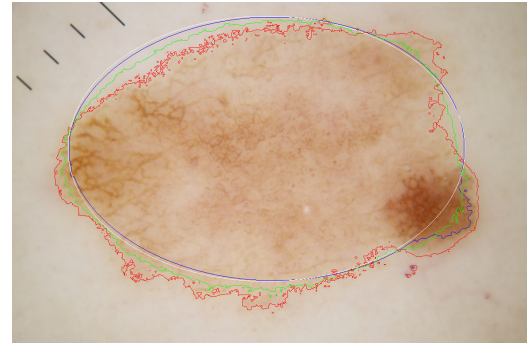
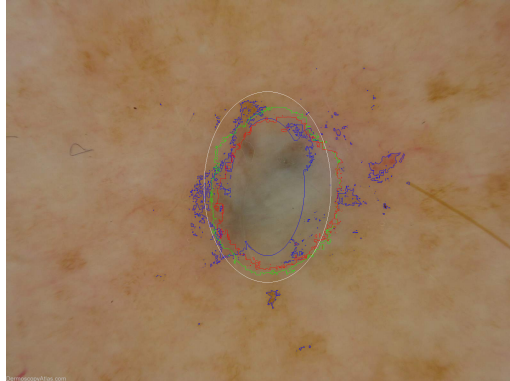
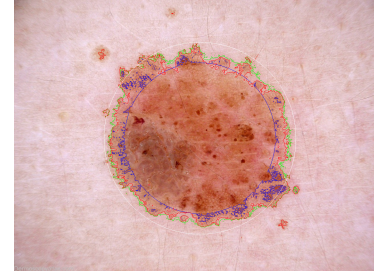
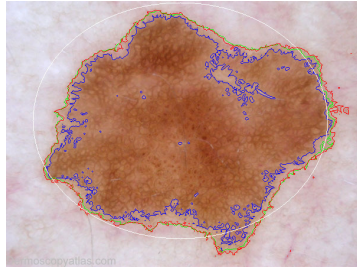


Figure 4.33: Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), and GS (blue) for D1- D7.



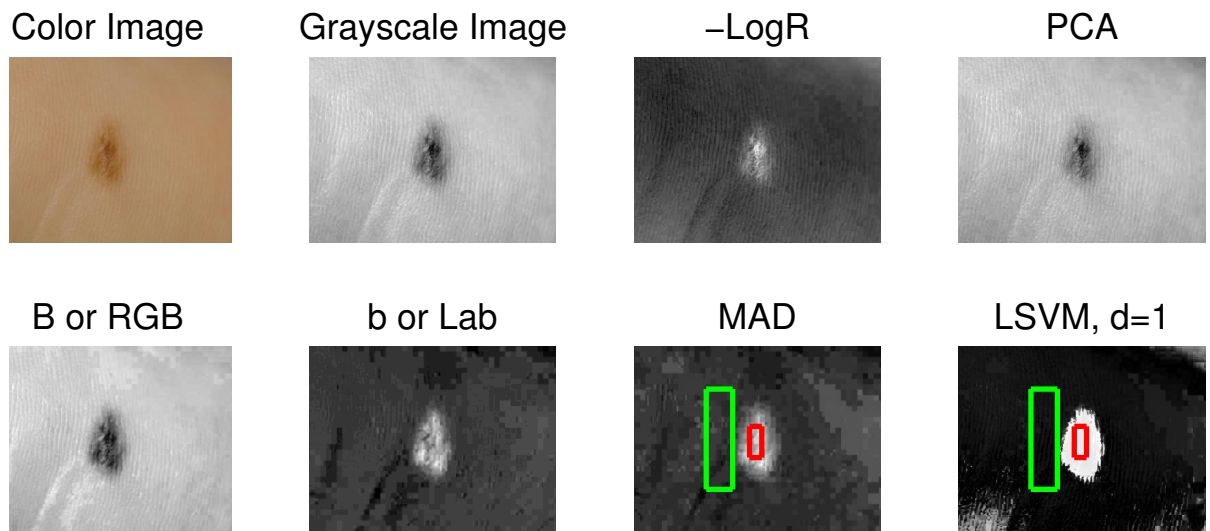


Figure 4.34: S1- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

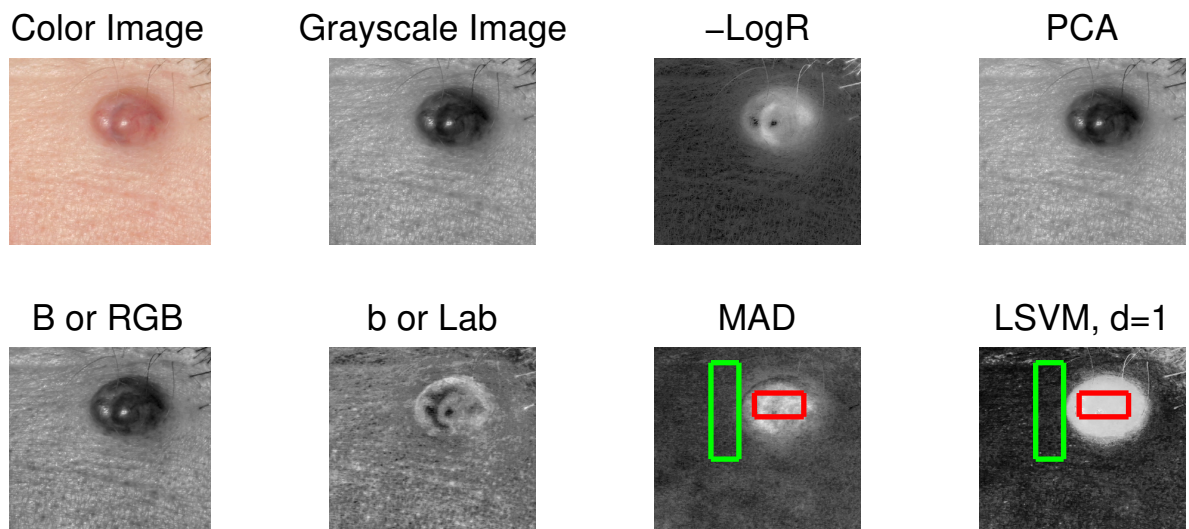


Figure 4.35: S2- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

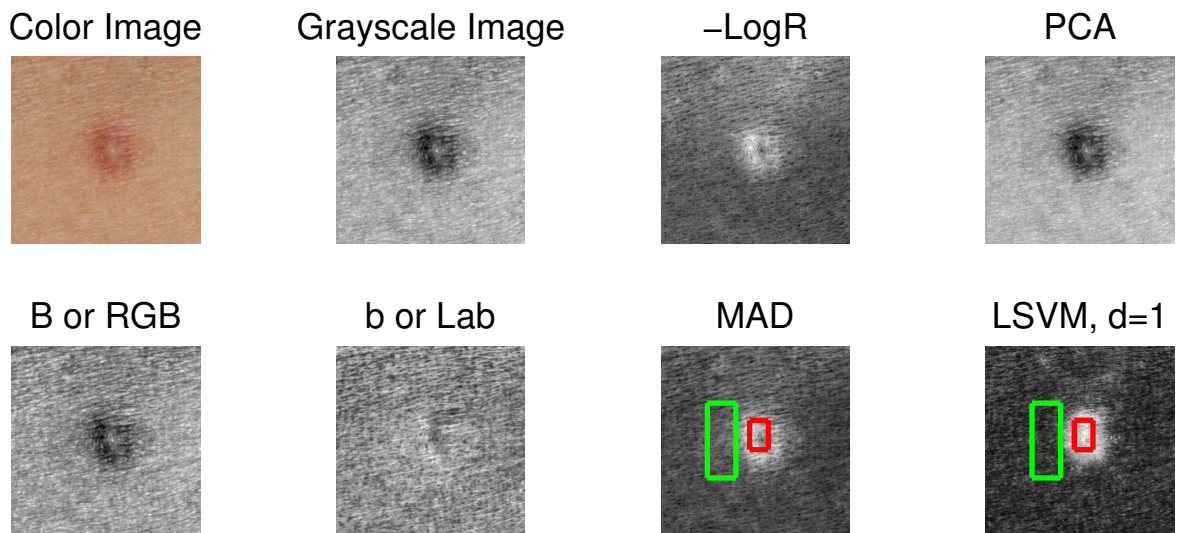


Figure 4.36: S3- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

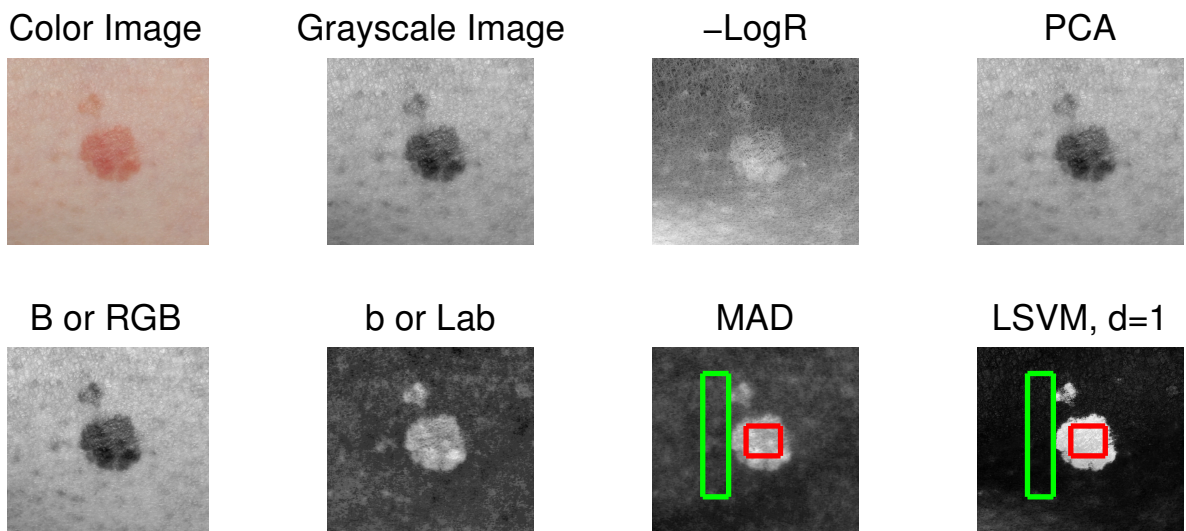


Figure 4.37: S4- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

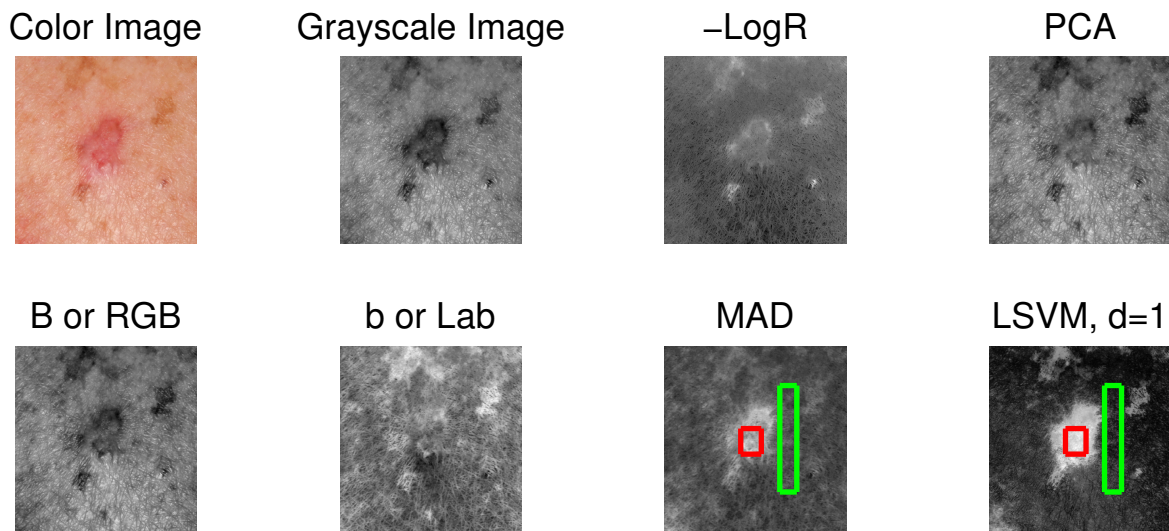


Figure 4.38: S5- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

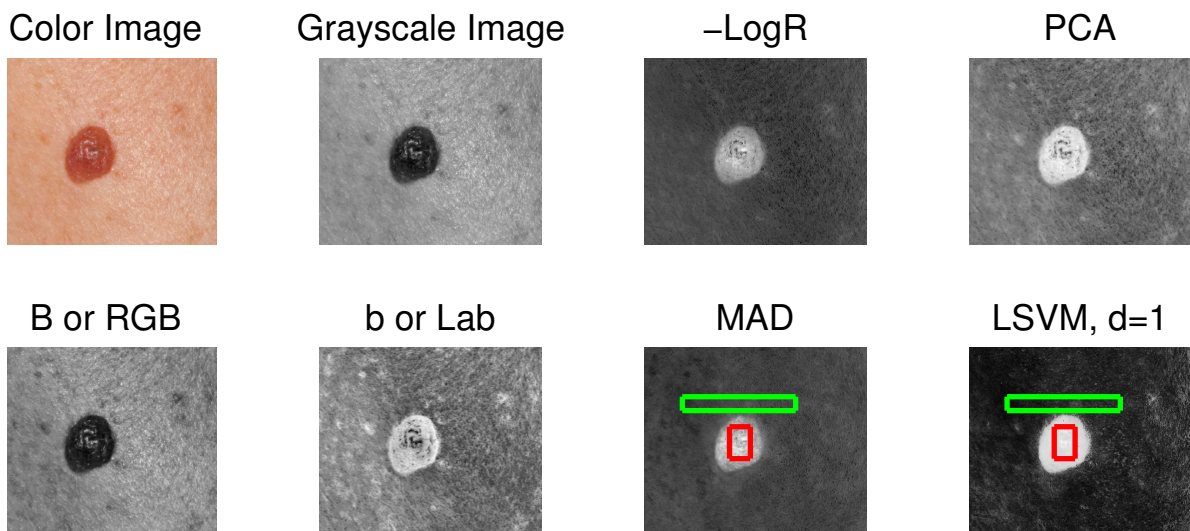


Figure 4.39: S6- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

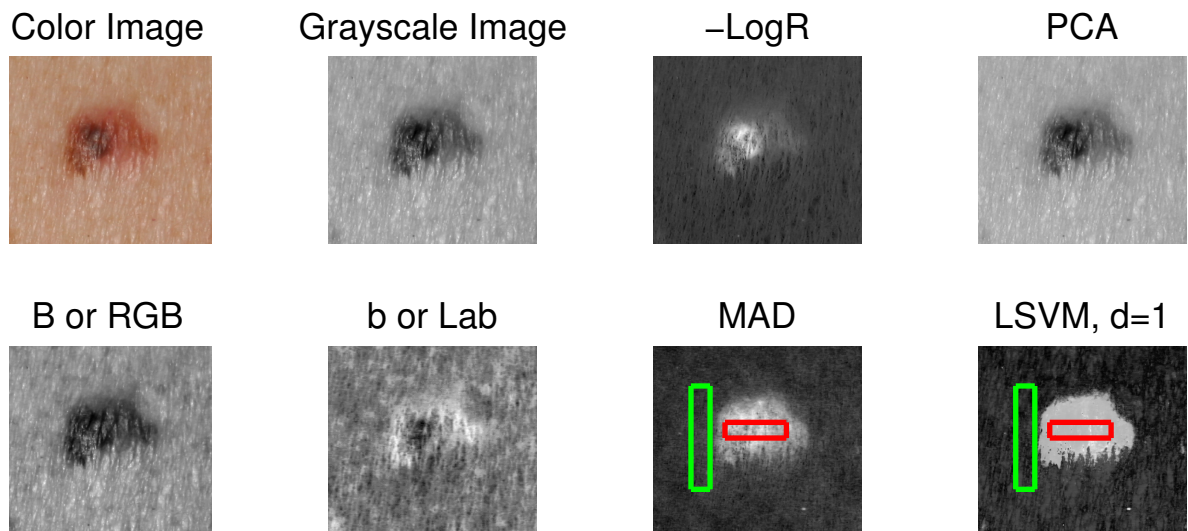


Figure 4.40: S7- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.

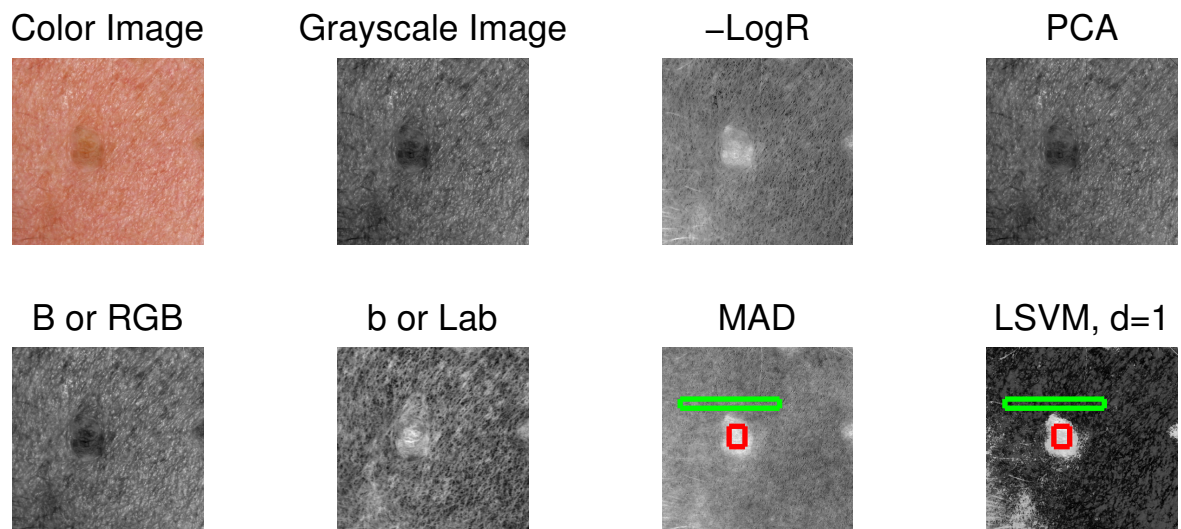


Figure 4.41: S8- Top: Color, GS, -LogR. Bottom: “B” of RGB, “b” of Lab, MAD, and LSVM.



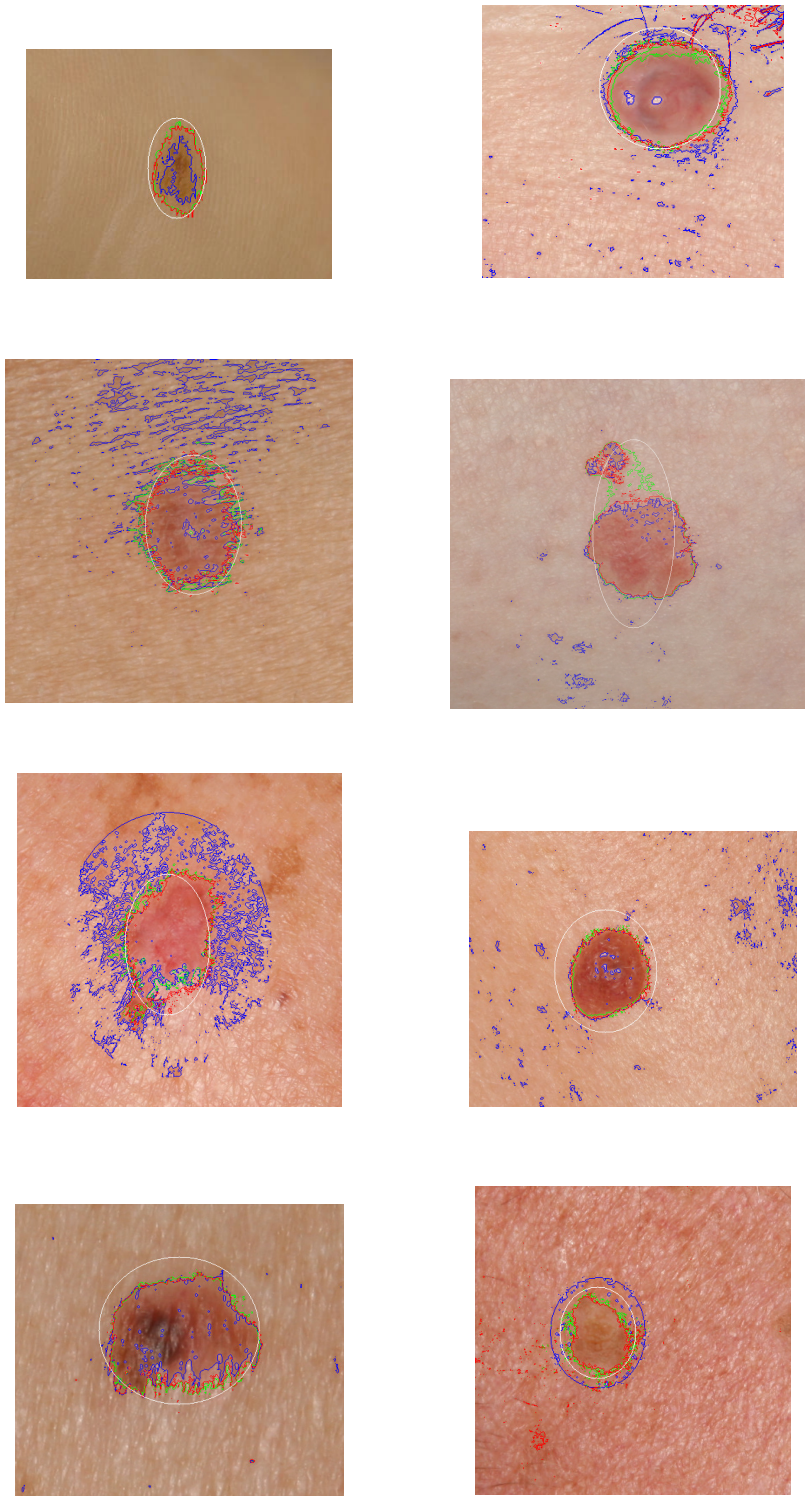


Figure 4.42: Resulting boundary after Chan-Vese on the Single-Coordinate Image. Starting contour (white), LSVM (red), MAD (green), and GS (blue) for S1- S8.

linear method and is significantly more time consuming to compute (Table 4.2.4, 4.2.5, 4.3.4 and 4.3.4). This suggests that it may not be robust enough to incorporate into an application or device. For these examples the linear DSGS method, for the most part, found boundaries that were consistent with the nonlinear DSGS method and far less dependent on user supplied information that can be inconsistent. This suggests that even when there is benefit from the nonlinear DSGS methods, the linear method may be sufficient, generating much better images than the traditional grayscale, for the given task.

Additional testing on larger datasets that include more variation in skin prototypes is needed. Although our data set for measuring erythema and wheals was small, we showed results over a variety of photo types (including darker skin tones). We demonstrated that the our new class of DSGS methods was much more consistent over a variety of skin tones then the previous suggested methods. With more testing we suspect that our method will be robust at detecting affected skin in darker skin tone (prototype IV–VI). Detecting erythema and lesions on darker skin tones is a major challenge for automated systems.

## Part II

### User Aided Parameter Selection for Image Deblurring

## Chapter 5: GIDE: A Tool for Graphical Image Deblurring Exploration

We continue the theme of finding user-aided solutions in image processing. In this part we introduce a tool that helps practitioners deblur a blurred image using their knowledge or intuition in a way that is safeguarded from possible bias.

Restoring a blurred image requires choice of a regularization method and associated parameter. Practitioners faced with these choices might favor results biased by what they expect to see and thereby introduce image artifacts or miss true image features. This is demonstrated in Figure 5.1. In this case the practitioner might not expect the moon to have train tracks and may favor a reconstruction where that information is lost such as the reconstructed image in the center of Figure 5.1. To avoid this, we developed a methodology for method choice and parameter selection that uses three statistical diagnostics to validate solutions, under the assumption of Gaussian additive noise in the blurred image. We packaged the methodology into the MATLAB software, called **GIDE**, which includes a user friendly graphical user interface (GUI). The software was built upon the existing **RestoreTools** package. It allows practitioners (or students) to visually explore the range of statistically likely solutions resulting from any of three regularization methods: Tikhonov, truncated



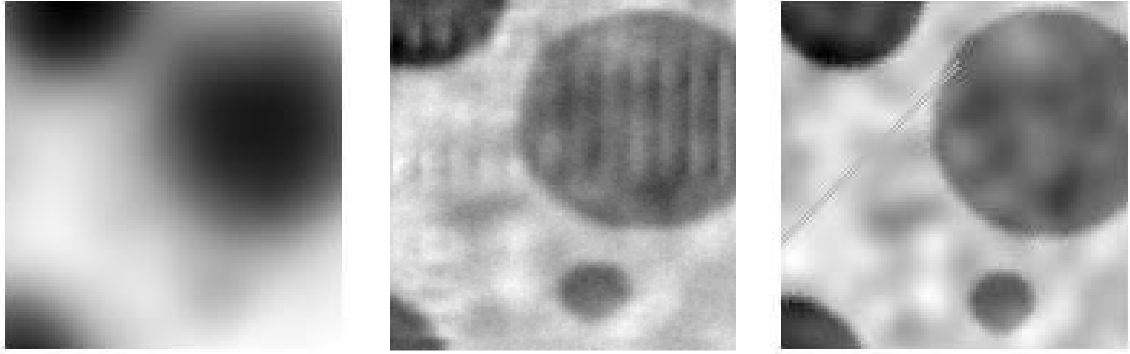


Figure 5.1: Left: Blurred Image, Center: Deblurred Image, Right: True Image which has “train tracks”. Without knowledge of the true image having “train tracks” one might accept the deblurred to be a good image without realizing that important information was lost in the process. Source: Dianne O’Leary.

SVD, and total variation.

## 5.1 Background

Medical and scientific imaging takes raw noisy data from a scientific instrument (MRI, CT, astronomical camera, etc.) and processes (“deblurs”) this data to produce images that are useful to practitioners. These rather expensive images are often critical in making medical or scientific decisions, so it is important that the deblurring is performed well. Unfortunately, image deblurring is an example of an ill-posed inverse problem, in which small changes (e.g., noise) in the data can make arbitrarily large changes in the deblurred image.

To overcome this limitation, we typically use our knowledge about the particular problem to formulate constraints [37]. For example, we might choose to limit the norm of the solution image (Tikhonov regularization), approximate the blurring operator by a low-rank matrix (Truncated Singular Value Decomposition (TSVD)

regularization), or limit the Total Variation (TV) of the solution. Adding these constraints replaces the ill-posed problem by one that is well-posed, and thus has a well-determined solution that we hope is near the true solution. Each of these methods requires the choice of a parameter we called  $\gamma$ .

In this work we take the viewpoint of a practitioner looking for a good reconstruction of a single image, or of a student gaining experience with various methods. Figure 5.2 shows that different regularization methods can yield very different reconstructions. Finding an appropriate method and parameter for a given problem is difficult, relying on properties of the particular problem and knowledge of the application area. Practitioners often have invaluable experience that is crucial in finding a good approximate solution, but too much reliance on intuition can lead them to see what is expected rather than the true solution. Any candidate reconstruction should be validated using statistical analysis.

We present a methodology and software with a Graphical User Interface (GUI) that can be used by practitioners to choose an appropriate regularization method and associated parameter while reducing the bias that can be introduced by searching for a visually appealing reconstruction. This methodology gives practitioners the ability to compare regularization methods by presenting a plausible range for  $\gamma$  and by presenting results of statistical tests of plausibility of each candidate image as a solution to the original ill-posed problem.

Our proof-of-concept software package **GIDE** (Graphical Image Deblurring Exploration) was built in MATLAB using the **RestoreTools** package [56]. Figure 5.3 shows a screen shot of the interface. A user can either upload a blurred image

or choose from samples provided. Similarly, a user provides a blurring matrix or a separable Point Spread Function (PSF), or chooses the default boxcar or Gaussian blurs. After selecting one of the regularization methods (TSVD regularization, Tikhonov regularization, or TV regularization), clicking “Compute” produces an initial solution based on automatic selection of the regularization parameter  $\gamma$ . The resulting deblurred image appears, along with the results of the statistical diagnostics. The user then uses the slide bar to adjust  $\gamma$ . This changes the resulting image and diagnostics in real time, allowing the user to explore the range of statistically plausible solutions.

In the remainder of this chapter we briefly review the deblurring problem in Section 5.2, the regularization methods in Section 5.3, initial parameter choices in Section 5.4, statistical diagnostics in Section 5.5, and software implementation in Section 5.6. Finally we discuss testing and validation in Section 5.7 and present a summary and conclusion in Section 5.8

## 5.2 Mathematical Model of Regularization Methods

An image can be thought of either as a real continuous function or as a collection of discrete square pixels.

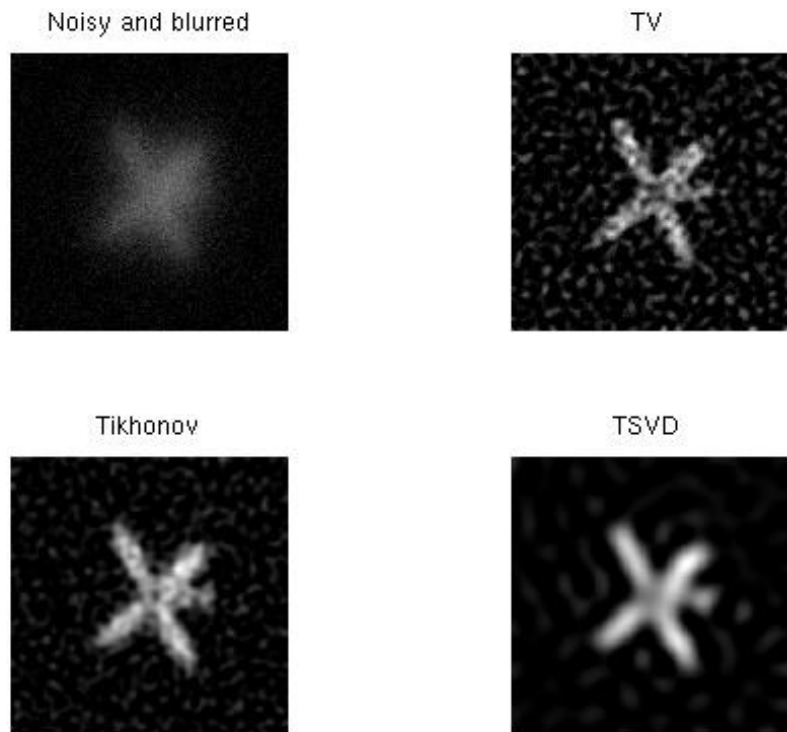


Figure 5.2: Restorations of  $256 \times 256$  blurred satellite image provided in `RestoreTools` with zero boundary conditions,  $\text{SNR}=9$  and parameters chosen by the initial parameter methods detailed in Section 5.4.

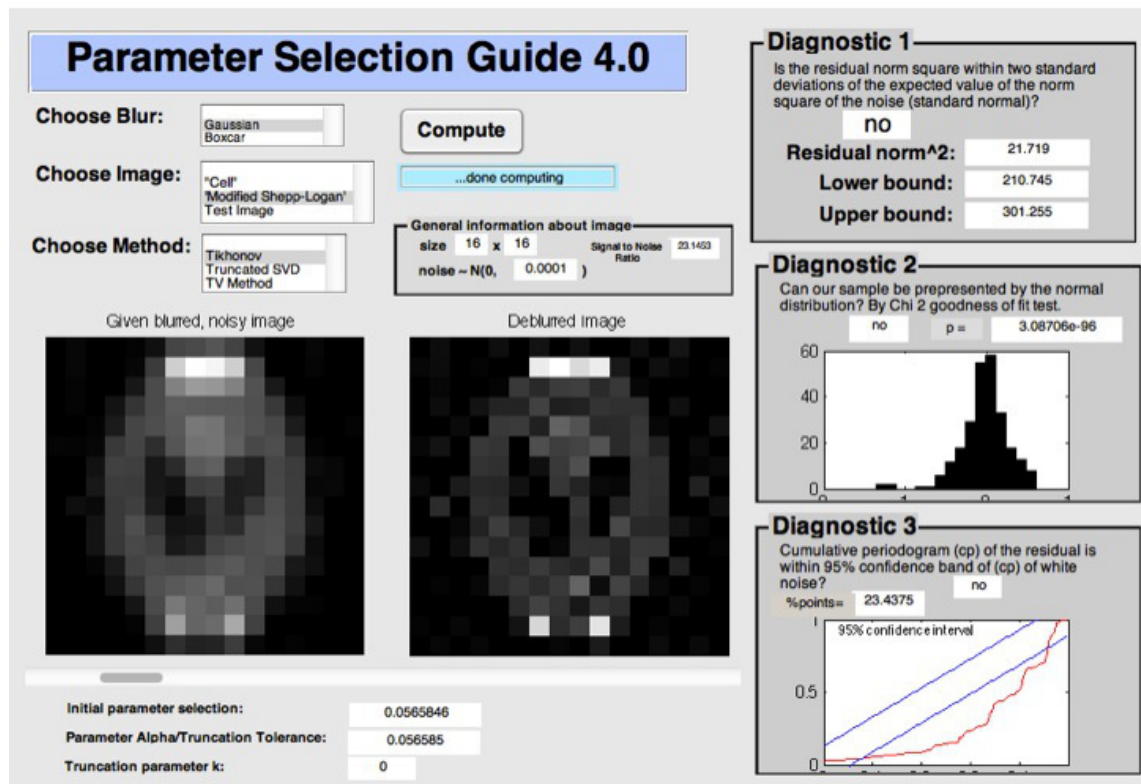


Figure 5.3: Final implementation of the GUI with the three diagnostics

### 5.2.1 Continuous Representation

In the continuous case, the blurring model can be represented by

$$Ku + \nu = u_0, \quad (5.1)$$

where  $K$  is the known blurring operator,  $u$  is the true image,  $u_0$  is the observed image, and  $\nu$  arises from a noise process that we assume to be Gaussian white noise process. Equation (5.1) is an example of an ill-posed problem since the operator  $K$  is an infinite-dimensional operator. Stability is imposed by adding a regularization or penalty term  $R(u)$  which incorporates *a priori* assumptions about the size and smoothness of the desired solution. Our problem becomes

$$\min_u \frac{1}{2} \|Ku - u_0\|_2^2 + \gamma R(u), \quad (5.2)$$

where  $\gamma$  is a nonnegative parameter.

### 5.2.2 Discrete Representation

We can model 2D deblurring problems either as determination of a real function of two variables or determination of a piecewise constant function, evaluated at discrete  $n_h \times n_v$  pixels, where  $h$  denotes horizontal and  $v$  denotes vertical. In the discrete case, a linear model of blur takes the form

$$\mathbf{A}\mathbf{x} + \boldsymbol{\epsilon} = \mathbf{b}, \quad (5.3)$$

where  $\mathbf{A}$  is a (known)  $m \times n$  blurring matrix with  $m \geq n$ ,  $\mathbf{x}$  is an (unknown)  $n \times 1$  vector containing pixel values of the true image, with  $n = n_h n_v$ ,  $\boldsymbol{\epsilon}$  is an (unknown)

$m \times 1$  vector of noise that we assume to be drawn from a normal distribution, and  $\mathbf{b}$  is the (known)  $m \times 1$  blurred and noisy image data, with  $m = m_h m_v$ . Equation (5.3) is a discrete ill-posed problem because  $\mathbf{A}$  is an ill-conditioned matrix approximating the infinite-dimensional blurring operator.

To regularize, we replace (5.3) by

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \gamma Q(\mathbf{x}). \quad (5.4)$$

The first term ensures fidelity to the model (5.3), while the function  $Q$  is chosen to assure that the minimization problem is well-posed. The scalar parameter  $\gamma$  is chosen to balance these two objectives.

### 5.2.3 Constructing the Blurring Matrix from the Point Spread Function

Consider an image of a single white pixel surrounded by black pixels. The image resulting from blurring this image is called the *point spread function (PSF)*. We construct the blurring matrix from this PSF, assuming a spatially invariant PSF.

For more details regarding finding the PSF and constructing the blurring matrix see the text [39].

## 5.3 Regularization Methods

GIDE gives the user a choice of three different regularization methods. Two of the methods are based on the Singular Value Decomposition (SVD) of  $\mathbf{A}$  (Tikhonov

and TSVD) and were chosen because of their popularity and ease of implementation. The third method, TV, was chosen because it favors solutions that include steep gradients (edges), typical of real images [87].

### 5.3.1 SVD-Based Regularization Methods

Define the SVD of  $\mathbf{A}$  to be

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (5.5)$$

where  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$  is an  $m \times n$  matrix and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$  is an  $n \times n$  matrix, each with orthonormal columns, and the diagonal matrix  $\mathbf{\Sigma}$  has entries  $\sigma_1 \geq \dots \geq \sigma_n \geq 0$  [38].

The Tikhonov regularization function is

$$Q_{\text{tik}}(\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|_2^2,$$

where  $\mathbf{L}$  is the identity matrix, an approximation of the first derivative operator, a diagonal weighting matrix [37, p. 12], or a problem-specific operator. We choose  $\mathbf{L}$  to be the identity matrix, and the Tikhonov solution is then given [38] by

$$\mathbf{x}_\gamma = \sum_{i=1}^n \frac{\sigma_i}{\sigma_i^2 + \gamma} (\mathbf{u}_i^T \mathbf{b}) \mathbf{v}_i. \quad (5.6)$$

In TSVD we regularize the problem by truncating  $\mathbf{A}$ . Effectively,  $Q_{\text{tsvd}}$  puts an infinite penalty on using components  $i$  of the SVD for which  $\sigma_i$  is too small. The TSVD solution is given in [38] by

$$\mathbf{x}_\gamma = \sum_{i=1}^n \frac{\phi_i}{\sigma_i} (\mathbf{u}_i^T \mathbf{b}) \mathbf{v}_i \quad (5.7)$$



where  $\phi_i = 1$  for  $i \leq \gamma$  and  $\phi_i = 0$  otherwise. In this case the regularization parameter  $\gamma$  is an integer chosen from the interval  $[1, n]$ .

### 5.3.2 TV Regularization for the Continuous Problem

Before discussing the implementation of TV regularization, we explain the concepts in terms of the continuous problem. In TV regularization, our regularization function, is the  $l_1$  norm of the gradient ( $\nabla$ ) of the solution and thereby retains sharp edges in the image that may be obscured if, for example, the  $l_2$  norm is used, as in Tikhonov regularization. The regularization function is

$$R(x) = \int_{\Omega} |\nabla x| d\Omega, \quad (5.8)$$

where

$$|\nabla x| = \sqrt{x_s^2 + x_t^2} \quad (5.9)$$

and  $\Omega$  is the domain of the image in the coordinate system  $(s, t)$ .

The continuous problem becomes the minimization of

$$f(u) = \frac{1}{2} \|Ku - u_0\|_2^2 + \gamma \int_{\Omega} |\nabla u| d\Omega. \quad (5.10)$$

The first-order condition of optimality (also known as the Euler-Lagrange equation) for the problem with homogeneous Neumann boundary conditions is

$$0 = K^*(Ku - u_0) - \gamma \nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right), \quad (5.11)$$

where  $K^*$  is the adjoint operator of  $K$  in the  $l_2$  inner product space. This regularization problem is non-linear and the TV term is not everywhere differentiable.

Often the difficulty of the  $TV(u)$  term not being differentiable at zero is avoided by adding a small positive constant  $\beta > 0$  so that (5.11) becomes

$$g(u) = K^*(Ku - u_0) - \gamma \nabla \cdot \left( \frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) = 0. \quad (5.12)$$

Unlike the Tikhonov and TSVD problems, which have the closed-form solutions (5.6) and (5.7), the TV problem must be solved iteratively by methods such as time marching schemes, Newton's method, lagged diffusivity fixed point iteration, or primal-dual Newton method. We used a primal-dual Newton method as presented in [11].

### 5.3.2.1 Formulation of Newton's Method

The first step of this method is to find a quadratic model that fits the function  $f(u + \delta u)$  where  $f(u)$  is given in (5.10), and  $\delta u$  is the search direction. The search direction can be found by minimizing the quadratic model over  $\delta u$ . The resulting step  $\delta u$  satisfies

$$H(u)\delta u = -g(u), \quad (5.13)$$

where

$$H(u) = K^*K - \gamma \nabla \cdot \left( \frac{1}{\sqrt{|\nabla u|^2 + \beta}} \left( I - \frac{\nabla u \nabla u^T}{|\nabla u|^2 + \beta} \right) \nabla \right). \quad (5.14)$$

### 5.3.2.2 Linearization Based on Introducing a New Variable

In [11] the authors suggest an improved method to solving the above problem by introducing a new variable

$$w = \frac{\nabla u}{|\nabla u|} \quad (5.15)$$

or

$$w = \frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}. \quad (5.16)$$

This technique is related to primal-dual optimization (details on the dual and primal-dual optimization problem can be found in Appendix 5.3.3.3) and gives better global convergence behavior than Newton's method [11].

From (5.12) we find the equivalent system of equations:

$$K^*(Ku - u_0) - \gamma \nabla \cdot w = 0, \quad (5.17)$$

$$w \sqrt{|\nabla u|^2 + \beta} - \nabla u = 0. \quad (5.18)$$

We now linearize this system to find:

$$\begin{bmatrix} \sqrt{|\nabla u|^2 + \beta} & -(I - \frac{w \nabla u^T}{\sqrt{|\nabla u|^2 + \beta}}) \nabla \\ -\gamma \nabla \cdot & K^* K \end{bmatrix} \begin{bmatrix} \delta w \\ \delta u \end{bmatrix} = - \begin{bmatrix} f(w, u) \\ g(w, u) \end{bmatrix}. \quad (5.19)$$

This method is called the primal-dual Newton's method [11].

### 5.3.3 Discretization of the TV Regularization Method

The method of Section 5.3.2 has a discrete analogue. After discretization, the regularization function becomes

$$Q_{\text{tv}}(\mathbf{x}) = \sum_{i=1}^n \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|_2^2 + \beta}. \quad (5.20)$$

Assuming  $\mathbf{x}$  is stacked by columns,  $\mathbf{D}_i^T \mathbf{x} = [x_{i+n_v} - x_i, x_{i+1} - x_i]^T$  is the discretization of  $\nabla x$  (with zero used when a subscript is out of range) [11]. A small  $\beta > 0$  is added because the TV term is not differentiable at zero. The resulting discrete TV problem

is

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^n \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|_2^2 + \beta}. \quad (5.21)$$

### 5.3.3.1 Discrete Formulation of Newton's Method

The discretization of (5.12) and (5.14) yields

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \gamma \mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T \mathbf{x}, \quad (5.22)$$

where

$$\nu_i = \sqrt{\|\mathbf{D}_i^T \mathbf{x}\|^2 + \beta}, \quad (5.23)$$

$$\mathbf{E} = \text{diag}(\nu_i \mathbf{I}_2)_{i=1, \dots, m}, \quad (5.24)$$

$\mathbf{I}_2$  is a  $2 \times 2$  identity matrix and

$$\mathbf{H}(\mathbf{x}) = \mathbf{A}^T \mathbf{A} + \gamma \mathbf{D}\mathbf{E}^{-1}\mathbf{F}\mathbf{D}^T, \quad (5.25)$$

where

$$\mathbf{F} = \text{diag}(\mathbf{I}_2 - \frac{\mathbf{D}_i^T \mathbf{x} \mathbf{x}^T \mathbf{D}_i}{\nu_i^2})_{i=1, \dots, m}. \quad (5.26)$$

### 5.3.3.2 Formulation of Primal-Dual Newton's Method

Analogous to the discretization in Section 5.3.3.1, we introduce the  $2m \times 1$  vector  $\mathbf{y}$  of dual variables and the discretization of the Primal-Dual Newton method becomes

$$\begin{bmatrix} \mathbf{E} & -\bar{\mathbf{F}}\mathbf{D}^T \\ \gamma\mathbf{D} & \mathbf{A}^T \mathbf{A} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{y} \\ \Delta \mathbf{x} \end{bmatrix} = - \begin{bmatrix} \mathbf{E}\mathbf{y} - \mathbf{D}^T \mathbf{x} \\ \gamma\mathbf{D}\mathbf{y} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \end{bmatrix}, \quad (5.27)$$

where  $\bar{\mathbf{F}} = \text{diag}(\mathbf{I}_2 - \frac{\mathbf{y}_i^T \mathbf{x}^T \mathbf{D}_i}{\nu_i})_{i=1,\dots,m}$ . This can be written as:

$$\mathbf{C}\Delta\mathbf{x} = \mathbf{g}(\bar{\mathbf{x}}), \quad (5.28)$$

$$\Delta\mathbf{y} = -\mathbf{y} + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\Delta\mathbf{x}, \quad (5.29)$$

where

$$\mathbf{C} = \gamma\mathbf{D}\mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T + \mathbf{A}^T\mathbf{A} \quad (5.30)$$

and

$$\mathbf{g}(\bar{\mathbf{x}}) = -(\gamma\mathbf{D}\mathbf{E}^{-1}\mathbf{D}^T\mathbf{x} + \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})). \quad (5.31)$$

Note that  $\mathbf{C}$  is not symmetric and as suggested by the authors of [11] should be replaced by symmetrization  $\bar{\mathbf{C}} = \frac{1}{2}(\mathbf{C} + \mathbf{C}^T)$  of  $\mathbf{C}$ .

### 5.3.3.3 Relationship to the Dual Problem

To understand the relationship of the above formulation to the primal-dual problem we first rewrite our minimization or primal problem (5.21) as a min-max problem by using the the following fact:  $\|\mathbf{a}\| = \max_{\|\mathbf{b}\| \leq 1} \mathbf{a}^T \mathbf{b}$  for any vector  $\mathbf{a}$ , as a result of the Cauchy-Schwartz inequality. Ignoring  $\beta$  we find that  $\|\mathbf{D}^T\mathbf{x}\| = \max_{\|\mathbf{y}_i\| \leq 1} \mathbf{x}^T \mathbf{D}\mathbf{y}$ . The resulting primal problem is the minimization with respect to  $\mathbf{x}$  of:

$$\mathbf{P}(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \gamma \max_{\|\mathbf{y}_i\| \leq 1} \mathbf{x}^T \mathbf{D}\mathbf{y}, \quad (5.32)$$

and from the concept of duality we know that the dual problem is simply the max-min problem [35, 57]. For more details on duality see Appendix A.

Our dual problem becomes the maximization with respect to  $\mathbf{y}$  of

$$\mathbf{N}(\mathbf{y}) = \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x}^T \mathbf{Dy}. \quad (5.33)$$

Because the function  $\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x}^T \mathbf{Dy}$  is convex in  $\mathbf{x}$  and concave (actually linear) and bounded for  $\|\mathbf{y}_i\| \leq 1$  we know there is a point that satisfies the saddle-point condition. This implies that there is strong duality so:

$$\min_{\mathbf{x}} \mathbf{P}(\mathbf{x}) = \max_{\|\mathbf{y}_i\| \leq 1} \mathbf{N}(\mathbf{y}). \quad (5.34)$$

The authors in [11] suggest considering the primal-dual problem, formulated by considering the conditions for which the difference between the primal and the dual objective function (duality gap) is zero, which holds when  $\mathbf{x}$  and  $\mathbf{y}$  are optimal:

$$\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^m \mathbf{y}_i^T \mathbf{D}_i \mathbf{x} = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^m \|\mathbf{D}_i^T \mathbf{x}\|, \quad (5.35)$$

where equality holds when  $\mathbf{y}_i^T \mathbf{D}_i^T \mathbf{x} = \|\mathbf{D}_i^T \mathbf{x}\|$  for every  $i$  where  $\mathbf{D}_i^T \mathbf{x} \neq 0$  and  $\|\mathbf{y}_i\| \leq 1$ . The primal-dual formulation becomes

$$\mathbf{A}^T \mathbf{Ax} - \mathbf{b} + \gamma \mathbf{Dy} = 0, \quad (5.36)$$

$$\|\mathbf{D}_i^T \mathbf{x}\| \mathbf{y}_i - \mathbf{D}_i^T \mathbf{x} = 0, \quad (5.37)$$

and

$$\|\mathbf{y}_i\| \leq 1. \quad (5.38)$$

We know the solution of the above system yields the optimal  $\mathbf{x}$  for the primal problem. Again we add a small constant  $\beta$  in order to make the above well defined for all  $i$ . Therefore our primal-dual formulation yields the discretization of (5.17) - (5.18).

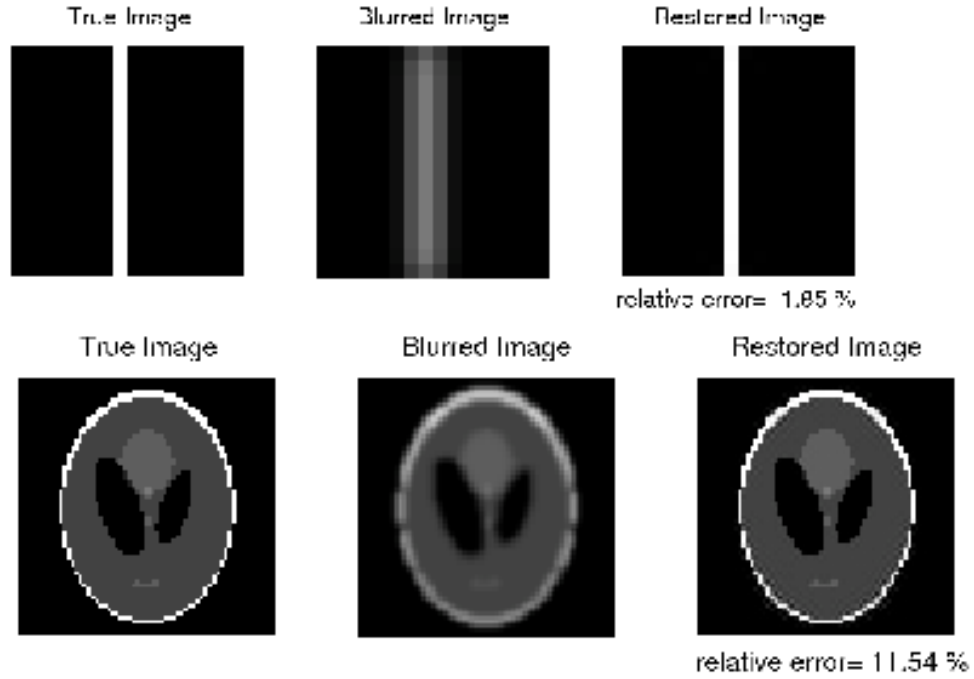


Figure 5.4: Results of Newton-CG for Total Variation regularization for two images. Top is a  $16 \times 16$  image generated in MATLAB and the bottom image is  $64 \times 64$  Modified Shepp-Logan generated in MATLAB.

#### 5.3.3.4 Implementing TV Regularization Method

The TV regularization method was implemented using the Primal-Dual Newton Method with Conjugate Gradients (CG) [11]. In Algorithm 1 and Algorithm 2 is the pseudo-code used for this work. In Figure 5.4 are results of applying the TV method on two test images.

---

**Algorithm 1** Pseudo-Code for Primal-Dual Newton Method

---

Initialize:  $\mathbf{x}_0$ , set  $k = 0$   
**while**  $\mathbf{x}_k$  is not “good enough” **do**  
    Solve  $\bar{\mathbf{C}}\Delta\mathbf{x}_k = -\bar{\mathbf{g}}(\mathbf{x})$  using CG to find the Newton direction  $\Delta\mathbf{x}_k$ .  
     $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\Delta\mathbf{x}_k$  where  $\alpha_k$  is determined by a linesearch<sup>1</sup>.  
     $\Delta\mathbf{y}_k = -\mathbf{y}_k + \mathbf{E}^{-1}\mathbf{D}^T\mathbf{x}_k + \mathbf{E}^{-1}\bar{\mathbf{F}}\mathbf{D}^T\Delta\mathbf{x}_k$ .  
     $\mathbf{y}_{k+1} = \mathbf{y}_k + s_k\Delta\mathbf{y}_k$  where  $s_k$  is determined by  $s_k = .9 \sup\{s_d : \|\mathbf{y}_i + s_d\Delta\mathbf{y}_i\| < 1, i = 1, \dots, m\}$ .  
    Set  $k = k + 1$   
**end while**  
“good enough”:  $\|\mathbf{g}(\mathbf{x}_k)\|/\|\mathbf{g}(\mathbf{x}_0)\| < 10^{-3}$ .

---

---

**Algorithm 2** Pseudo-Code CG

---

Initialize:  $\mathbf{r} = -\bar{\mathbf{g}}(\mathbf{x})_k - \bar{\mathbf{C}}(\mathbf{x})\Delta\mathbf{x}$ ,  $\mathbf{q} = \mathbf{r}$ ,  $\rho = \|\mathbf{r}\|$ ,  $\gamma = \rho^2$ .  
**while**  $\|\mathbf{r}\|/\rho \geq tol$  **do**  
     $\alpha = \frac{\gamma}{\mathbf{q}^T\bar{\mathbf{C}}(\mathbf{x})\mathbf{q}}$ .  
     $\Delta\mathbf{x} = \Delta\mathbf{x} + \alpha\mathbf{q}$ .  
     $\mathbf{r} = \mathbf{r} - \alpha\bar{\mathbf{C}}(\mathbf{x})\mathbf{q}$ .  
     $\hat{\gamma} = \|\mathbf{r}\|^2$ .  
     $\beta = \frac{\hat{\gamma}}{\gamma}$ ,  $\gamma = \hat{\gamma}$ .  
     $\mathbf{q} = \mathbf{r} + \beta\mathbf{q}$ .  
**end while**  
Tolerance for CG:  $tol = 0.1$  when  $k = 0$  and  $tol = \min(0.1, 0.9\|\mathbf{g}(\mathbf{x}_k)\|^2/\|\mathbf{g}(\mathbf{x}_{k-1})\|^2)$  are the suggested conditions in [11].

---



## 5.4 Initial Parameter Selection

A number of automated parameter selection methods have been developed, some based on prior knowledge of the particular problem (distribution of noise or errors), others based on statistical criteria. The parameters chosen by these methods are often far from those that minimize the deviation of the computed solution from the true solution [37]. In this work, automated parameter selection methods are used only to find a candidate range of parameters that the user can explore. We choose to use generalized cross validation (GCV) for the SVD-based methods, since the computation can be performed efficiently, and the discrepancy principle for TV, since GCV is too costly.

### 5.4.1 Generalized Cross-Validation (GCV)

GCV is based on the popular leave-one-measurement-out model, checking the reasonableness of a parameter determined from  $m - 1$  measurements by seeing how well the resulting model predicts the  $m$ th measurement [38, p. 95]. The idea is to choose the parameter  $\gamma$  that minimizes the prediction errors. In GCV we formulate this as minimizing

$$G(\gamma) = \sum_{k=1}^m [\mathbf{b}_k - (\mathbf{A}\mathbf{x}_{\gamma}^{(k)})_k]^2, \quad (5.39)$$

where  $\mathbf{x}_{\gamma}^{(k)}$  is the estimate that results from using the regularization parameter  $\gamma$  but omitting the  $k^{th}$  measurement of  $\mathbf{b}$ . For SVD-based methods, the GCV expression

---

<sup>1</sup>We used *cvsrch.m* which was original developed by Jorge J. More' and David J. Thuente as part of minpack and translated to MATLAB by Dianne O'Leary.

can be greatly simplified [39]. For Tikhonov regularization,  $G$  becomes

$$G_{\text{tik}}(\gamma) = \frac{\sum_{i=1}^m (\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i^2 + \gamma^2})^2}{\sum_{i=1}^m (\frac{1}{\sigma_i^2 + \gamma^2})^2} \quad (5.40)$$

while for TSVD,

$$G_{\text{tsvd}}(\gamma) = \frac{1}{(m - \gamma)^2} \sum_{i=\gamma+1}^m (\mathbf{u}_i^T \mathbf{b})^2. \quad (5.41)$$

### 5.4.2 Discrepancy Principle

If we know the distribution of the noise  $\boldsymbol{\epsilon}$  then we can choose  $\gamma$  so that

$$\|\mathbf{A}\mathbf{x}_\gamma - \mathbf{b}\|_2 = \nu \mathcal{E}(\|\boldsymbol{\epsilon}\|_2), \quad (5.42)$$

where  $\mathcal{E}$  denotes expected value and  $\nu = 2$  is a safety factor [38, p. 90]. The appropriate value of  $\gamma$  is computed using an efficient root finding algorithm.

## 5.5 Statistical Diagnostics

We use statistical diagnostics to test the plausibility of a candidate regularization solution as a solution to the original ill-posed problem. We use the three diagnostics from [73] to generate a range of plausible regularization parameters.

These diagnostics are based on the simple observation that since

$$\boldsymbol{\epsilon} = \mathbf{b} - \mathbf{A}\mathbf{x} \quad (5.43)$$

is noise drawn from some statistical distribution, then

$$\mathbf{r}_\gamma = \mathbf{b} - \mathbf{A}\mathbf{x}_\gamma \quad (5.44)$$

should ideally equal  $\epsilon$  and therefore be a sample from the same distribution. We use standard statistical tests to evaluate how typical  $\mathbf{r}_\gamma$  is as a sample from the distribution, which we assume to be normal with known variance.

### 5.5.1 Choice of Diagnostics

To use statistical diagnostics, we renormalize our problem so that  $\epsilon \sim N(\mathbf{0}, \mathbf{I}_m)$ .

We now discuss the three diagnostics shown on the right side of the GUI in Figure 5.3.

#### 5.5.1.1 Residual Diagnostic 1

The sum of squares of a set of  $m$  independent identically distributed (i.i.d.) standard normal ( $\epsilon \sim N(\mathbf{0}, \mathbf{I}_m)$ ) random samples is a random variable with a  $\chi^2$  distribution. It has expected value  $m$  and variance  $2m$  [70]. Therefore, our first diagnostic tests whether the residual norm squared,  $\|\mathbf{r}_\gamma\|_2^2$ , should be within two standard deviations (i.e., within the 95% confidence interval) of the expected value of  $\|\epsilon\|_2^2$ . Therefore, we want

$$\|\mathbf{r}_\gamma\|_2^2 \in [m - 2\sqrt{2m}, m + 2\sqrt{2m}]. \quad (5.45)$$

GIDE displays the residual norm-squared and the endpoints of the confidence interval.

### 5.5.1.2 Residual Diagnostic 2

The histogram of the elements of  $\mathbf{r}_\gamma$  should look like a bell-shaped curve. **GIDE** displays the histogram and the result of a goodness-of-fit test [70]. We used a  $\chi^2$  goodness-of-fit test which tests whether the residual is drawn from an i.i.d standard normal distribution (null hypothesis) by comparing it to the theoretical distribution. **GIDE** displays the histogram of the residual as well as the *p-value* where if  $p > 0.05$  then one should accept the null hypothesis with 95% confidence.

### 5.5.1.3 Residual Diagnostic 3

If we view the elements of  $\boldsymbol{\epsilon}$  and  $\mathbf{r}$  as time series with index  $i = 1, \dots, m$  then  $\epsilon_i \sim N(0, 1)$  forms a white noise series. We would then expect  $\{r_i\}$  to also be a white noise series. One way to assess this is to compute the *cumulative periodogram* of the residual [28, Chapter 7]. First we compute the Fourier transform

$$r_t = \frac{a_o}{2} + \sum_{k=1}^p (a_k \cos(w_k t) + b_k \sin(w_k t)), \quad t = 1, \dots, m, \quad (5.46)$$

where  $p = \text{ceil}(\frac{m-1}{2})$  and the frequency is  $w_k = \frac{2\pi k}{p}$ . The periodogram is defined as

$$g(w_k) = \frac{p}{2} (a_k^2 + b_k^2), \quad k = 0, \dots, p \quad (5.47)$$

which represents the sum of squares of the coefficients at each frequency  $w_k$ . The cumulative periodogram is given by

$$c_j = \frac{\sum_{(k=0)}^j g(w_k)}{\sum_{(k=0)}^p g(w_k)}, \quad j = 0, \dots, p \quad (5.48)$$

If we look at the periodogram of white noise, the expected value of  $g(w_k)$  at every frequency is constant and a plot of  $g(w_k)$  should resemble a straight line from  $(0, 0)$  to  $(p, 1)$ . The 95% confidence interval for the plot of the cumulative periodogram of white noise is shown in `GIDE`, along with the plot of  $c_1, \dots, c_p$ . The endpoints of the interval are at approximately at plus or minus  $1.36/\sqrt{(p-1)}$  relative to the straight line when  $p > 31$  [28].

### 5.5.2 Validation of Residual Diagnostics

The three diagnostics all test the hypothesis that the residual components are drawn from a normal distribution, but the diagnostics are sensitive to somewhat different perturbations, as we illustrate here. Each of the diagnostics was tested first with standard normal i.i.d. samples to confirm that it is satisfied approximately 95% of the time; see Table 5.5.2. From the first line in the table, for 1000 different standard normal i.i.d. samples, this is true. In the second line of the table, we show the results of 1000 trials of perturbing every 100th sample by one. Diagnostics 1 and 2 accurately detect that the result is not normal, but Diagnostic 3 is only correct in 54% of the trials. In the third line, we add a small amount of Poisson-distributed noise; only Diagnostic 2 reports non-normality reliably.

Diagnostic 2 is reliable, but the others provide useful information as well. Diagnostic 1 gives the user a range of residual norms that satisfy that diagnostic; this helps the user easily find a range of feasible values for  $\gamma$ . Diagnostic 3 gives a useful visualization of how similar the residual is to the Gaussian distribution,

Table 5.1: For 1000 runs, number of times the Diagnostics fail to be satisfied.  $I(i) = 1$  if  $(i - 1) \bmod(100) = 0$  or  $(i - 2) \bmod(100) = 0$  and  $I(i) = 0$  otherwise.

Residual	Diag. 1	Diag. 2	Diag. 3
$r_i \sim N(0, 1)$	51	46	14
$r_i + I(i)$	950	999	539
$r_i + .05 * s_p \quad s_p \sim pois(1)$	156	1000	149

which is sometimes difficult to see from Diagnostic 2.

## 5.6 The Software

GIDE was built using MATLAB’s GUI toolbox. It was designed to enable a user with limited or no experience with the methods discussed to find suitable solutions to a problem. As Figure 5.3 shows, a user can select the type of blur, an example test problem, and the regularization method. Clicking “Compute” finds an initial solution based on the approximate parameter (found using GCV or the discrepancy principle). The user can then use the slide bar below the images to vary the regularization parameter in order to explore the results and find solutions that satisfy each of the three diagnostics on the right. The user has the option of using one of the pre-loaded blurs to generate a test problem (requires a true image) or to upload a blurring matrix for their own test problem (does not require knowledge of the true image). The blurring matrix titled “Boxcar” models a boxcar blur of band size three where the PSF is given by

$$PSF_{boxcar} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

The blurring matrix titled “Gaussian” models a Gaussian blur of band size three and  $\sigma = .7$  where

$$PSF_{gaussian} = \frac{1}{2\pi\sigma^2} \begin{pmatrix} \exp(-\frac{(-1)^2+1^2}{2\sigma^2}) & \exp(-\frac{0^2+1^2}{2\sigma^2}) & \exp(-\frac{1^2+1^2}{2\sigma^2}) \\ \exp(-\frac{(-1)^2+0^2}{2\sigma^2}) & \exp(-\frac{0^2+0^2}{2\sigma^2}) & \exp(-\frac{1^2+0^2}{2\sigma^2}) \\ \exp(-\frac{(-1)^2+(-1)^2}{2\sigma^2}) & \exp(-\frac{0^2+(-1)^2}{2\sigma^2}) & \exp(-\frac{(-1)^2+(-1)^2}{2\sigma^2}) \end{pmatrix}.$$

The SVD-based methods, the matrix structures, and the implementations of GCV and the discrepancy principle were taken from **RestoreTools** [56]. We implemented the primal-dual TV algorithm and the statistical diagnostics.

The speed of today’s computers limits the size of images for which real-time response is reasonable in the GUI. Nevertheless, it is quite practical to use the GUI to explore results for a subimage of an image of interest, in order to choose an appropriate regularization method and a statistically-validated candidate parameter for the full image.

GIDE is a working proof-of-concept that can be scaled to a faster computational tool by using a compiled computer language and high-performance computing.

## 5.7 Results and Testing

In this section we discuss some testing we performed on different components of GIDE on a variety of test images.

### 5.7.1 Test Images

We used artificially generated images and PSF functions for development and initial testing of the software. These data sets were created to be of any size. For

testing and validation we used the images found in *RestoreTools* as well as a variety of PSFs also found in *RestoreTools*.

### 5.7.2 Signal-to-Noise Ratio Effect on Diagnostics

As one may expect, the diagnostics are affected by the Signal-to-Noise Ratio (SNR) which is defined as

$$SNR = 10 \log_{10} \left( \frac{\|\mathbf{b}\|^2}{\|\epsilon\|^2} \right). \quad (5.49)$$

The range of plausible parameters that meet the diagnostics increases as the SNR approaches to zero and as the SNR increases the range of plausible solutions becomes smaller (Figure 5.5). Note that users should be careful when they have a very small SNR as the diagnostics used may not be the best measure for plausible solutions. This test was performed on a  $16 \times 16$  piece of Matlab test image “*cell.tif*” where the range of SNR was varied from 20 to 75. Ranges plotted are for Diagnostic 1 and the Tikhonov method, although a similar relationship was found for the different methods, diagnostics, and different test images included in the GUI.

### 5.7.3 Effects of $\gamma$ on Computation Time

We found that the computation time of the TV regularization method (dependent on the number of CG iterations) is dependent on the value of  $\gamma$ . See Figure 5.6 for results for different values of  $\gamma$ .



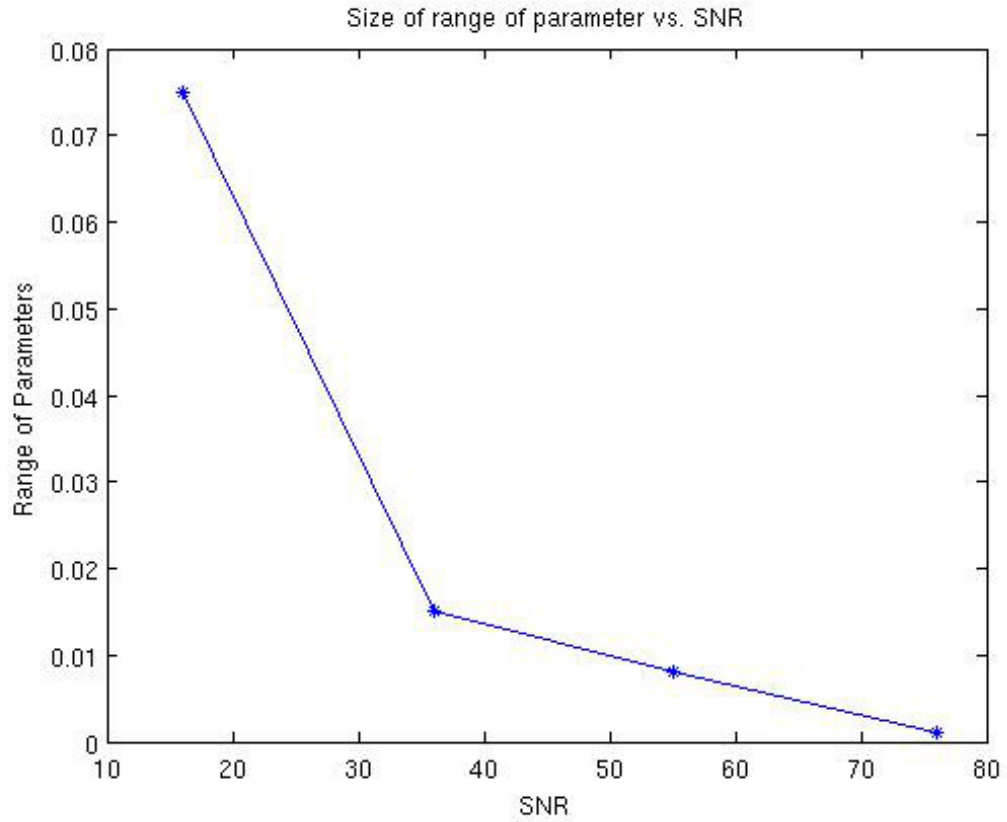


Figure 5.5: For the  $16 \times 16$  segment of the image “*cell.tif*” the range ( $\pm 0.0025$ ) was found for Diagnostic 1 using Tikhonov regularization. From the plot we can see that as the Signal-to-Noise Ratio decreases the range of parameters satisfying the diagnostic increases.

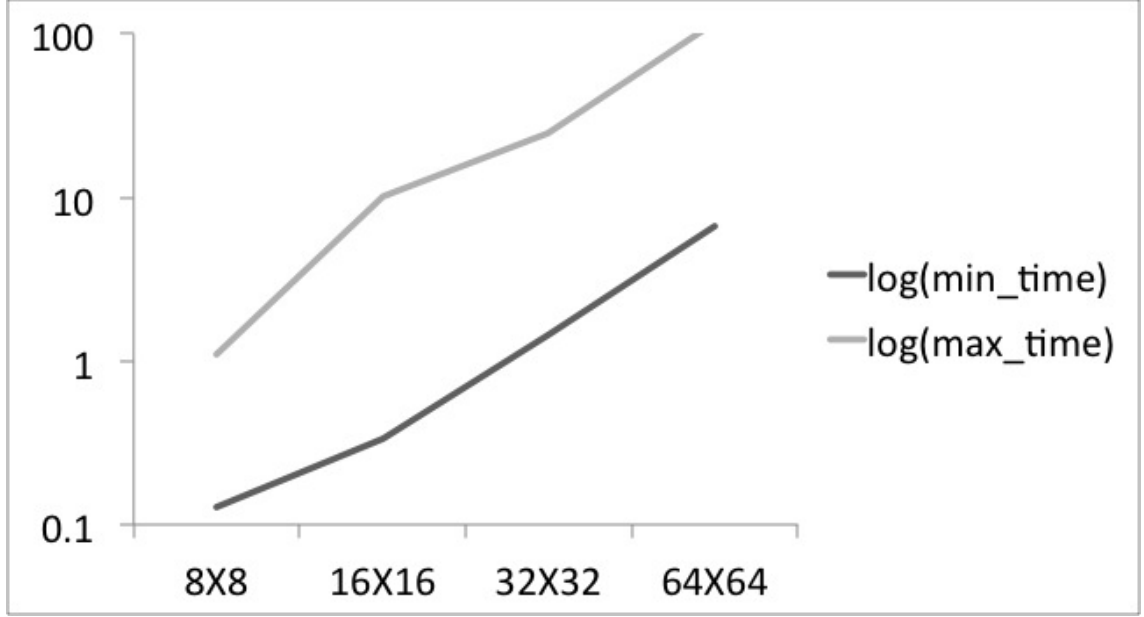


Figure 5.6: The difference in  $\log_{10}$  of computational time (seconds) for the TV regularization method for parameters between  $\gamma = 1$  and  $\gamma = 10^{-9}$ . For all image sizes the maximum time was for  $\gamma = 1$ .

As a result of this finding, we looked into using preconditioners to speed up the algorithm for large  $\gamma$ . An incomplete LU factorization (ILU) is suggested as a preconditioner for the Primal-Dual Newton method [11]. We did not find the use of *ILU* to be robust enough to be included in the GUI as we had to adjust the drop tolerance depending on  $\gamma$ .

#### 5.7.4 Results on Larger Images and Varied PSF

Although the GUI is only able to handle very small images, additional tests of the Primal Dual TV method were done on larger images as well as images with a variety of PSFs. See Figure 5.2 and Figure 5.7 comparing the results using Tikhonov and TSVD to the Primal Dual TV method.

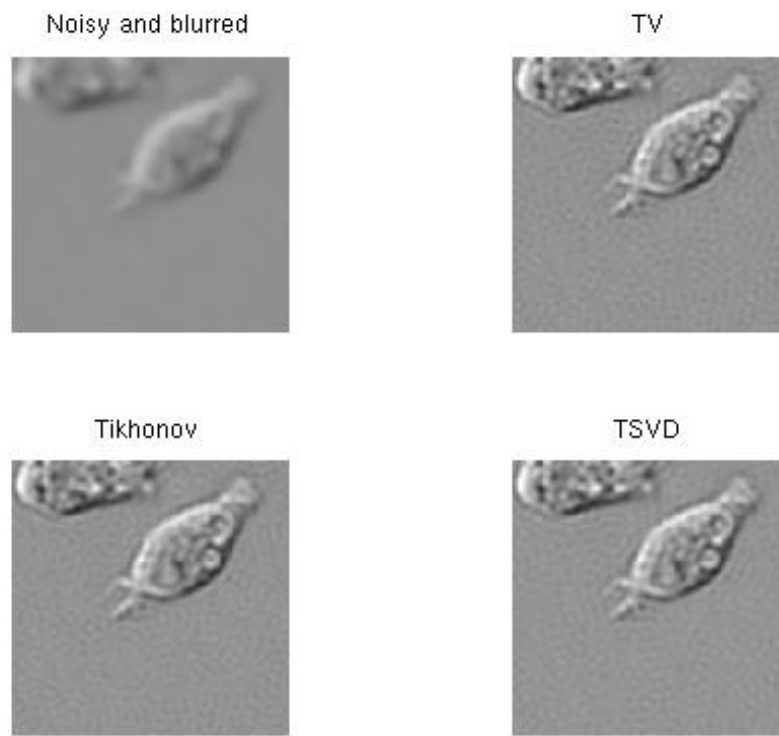


Figure 5.7:  $129 \times 129$  image of “*cell.tif*” with Gaussian blur and zero boundary conditions with SNR of 60.

## 5.8 Summary and Conclusions

When decisions are based on images, it is important to use a regularization method and parameter that can be justified on statistical grounds. In this work we present software to help practitioners do this. The software takes advantage of the practitioner's trained eyes while limiting bias by using statistical diagnostics. Without knowledge of the numerical method the user can explore different solutions with real time diagnostics determining whether the solution is statistically feasible.

There has been work in automatic parameter selection but these methods have shown not to be reliable over a variety of problems. Given this challenge and the lack of automatic approaches for choosing a regularization methods, our methodology is a straightforward approach for finding the appropriate method and parameter.

To effectively be used in real time, our methodology is currently limited to relatively small images. There remain many ways to scale the package as discussed in future work in Chapter 6. That being said the software has been proved useful in an undergraduate course on image restoration, giving the students immediate feedback about the effects of different regularization methods and parameter choices.

## Part III

### Conclusions and Future Work

## Chapter 6: Conclusions and Future Work

With the increasing availability of high resolution imaging tools, even in our pockets (i.e. smart phones), everyday users can do far more than simply digitally capturing a family moment. The ease of new applications available in these portable forms, linked with users who have expert knowledge about the images and task, opens the door to new possibilities. With this in mind we developed two new approaches that utilize the user’s knowledge for improved results. We applied these approaches to real life problems in medical and scientific image applications.

In the first approach, presented in Part I, we introduced a class of linear and nonlinear methods which we call the Domain-Specific Grayscale (DSGS) methods. A DSGS method transforms a color image into an image analogous to a grayscale image which utilizes user-specified information that is used to optimize the image processing task at hand and reduce the computations. We studied an application for which the DSGS methods proved to be an ideal method: DSGS proved useful for tasks of segmenting and finding the boundaries of skin lesions including skin cancer and responses to allergy tests.

In the second approach, presented in Part II, we introduced a methodology for improved image deblurring that safeguards against bias that can easily be introduced

as the user favors a particular reconstruction or deblurred image. This is particularly important in scientific and medical applications used for discovery or diagnosis.

## 6.1 Part I: Domain-Specific Grayscale For Improved Image Processing

In Chapter 2 and Chapter 3 we introduced a new class of methods, DSGS methods, that can greatly improve the segmentation of a color image versus the traditional approaches of using a grayscale image (as well as other suggested transformations) given knowledge of the background and foreground. Even in experiments where the background and foreground are multi-colored, we still found a space that enables us to pick out the object(s) of interest by utilizing nonlinear transformations. We saw that these methods were especially applicable to dermatological applications.

In Chapter 4 we applied DSGS to dermatological applications, and showed that the DSGS images had heightened contrast between the lesions and unaffected skin over previous suggested methods. In the case of measuring erythema and wheals after skin-prick test in allergy testing, we foresee the DSGS image as an important first step in developing an application for a smartphone for automatic measuring of erythema and wheals for allergy diagnosis. Such an application could be used in a variety of settings and eliminate the inconsistency between human measurements. Development of such an application would require the study of more data sets, development of protocols for best image capture, and clinical trials to determine

effectiveness and usefulness.

In the task of finding the boundary of possible cancerous lesions, we feel that the presented class of methods could improve existing algorithms' lesion classification as well as reduce the computational cost of some of these algorithms that utilize color images, making the software required more portable.

### 6.1.1 Key Contribution

- We introduced a new class of methods call Domain-Specific Grayscale (DSGS) methods which utilize user-specified information about the background and object(s) of interest in a color image to find a single-coordinate image that optimizes the contrast between object and background in order to improve results of edge detection and segmentation of that object. This new transformed image is a single coordinate image, like a grayscale image, and therefore is faster to process.
- We showed that the SVM method used for classification can be used as an effective DSGS method where the SVM score becomes the coordinate of the DSGS image. The use of SVM gives a natural extension to nonlinear methods through the use of kernels.
- We developed the new Minimizing Area between Distributions (MAD) which finds the orientation in **RGB** space that minimizes the overlapping area of the distribution of the selected foreground and background pixels if we assume they are normal distributed. This method was shown to be as effective as linear



SVM without the need for any user-specified parameters.

- We developed a new nonlinear method, the Gaussian Sum (GSUM) method, that showed potential of learning the background/foreground gamut. This is a challenging problem since the enclosing shape is not guaranteed to be convex or connected.
- We demonstrated that the choice of the log kernel in SVM is quite effective for problems involving color image data.
- We applied our DSGS methods in dermatological image analysis. We showed that in this application, the DSGS image had greater contrast between the affected skin and the background skin. This improvement over existing methods can potentially lead to improved classification of melanoma and other skin cancers and lead to an automatic application for detecting and determining a positive response to a skin-prick test and skin-patch test. For the task of measuring the wheal after a skin-prick test, our DSGS methods found edges that retained the general shape of the wheal versus human drawn edges. This suggests that the DSGS may be applicable for an automated diagnosis of a positive skin-prick test.

### 6.1.2 Future Work

- Develop a methodology to determine if the linear DSGS is suitable or if nonlinear DSGS method is warranted.
- In the case where a nonlinear method is necessary, develop a methodology for

selecting the kernel and kernel parameter (if a kernel method) based on the image application.

- Find a model to determine the probabilistic output of the foreground and background after SVM for further improved segmentation/edge detection. With this additional information standard image processing tools can be used for possible further improved tasks.
- Learn the shapes/gamut of the background and foreground using a method that is less dependent on variation in the data or that is better tuned to the data. Finding shapes that are not convex or even connected is not trivial. Previous work in compressed sensing and 3D graphics may be useful.
- Test the DSGS methods on a larger skin-prick data set to test the viability. This dataset should be taken under all sorts of conditions to test the robustness and to determine ideal conditions for capturing the image for most accurate results.
- Develop a smart phone application to measure wheal as a proof of concept and ultimately move to a clinical trial which would determine the feasibility of an "app" for detecting erythema and measuring wheals as a medical device.
- Apply the DSGS methods to hyperspectral images, including possibly finding a three coordinate false-color image to help aid the viewer.

## 6.2 Part II: A Tool for Graphical Image Deblurring Exploration (GIDE)

Our work created a methodology and software with a graphical user interface (GUI) that can be used by practitioners to choose an appropriate regularization method and associated parameter. At the same time it reduced the bias that can be introduced by searching for a visually appealing reconstruction. This methodology gives practitioners the ability to compare regularization methods while safeguarding them against bias by displaying statistical diagnostics on the residuals produced by various methods and parameter choices.

Mainstream research in image deblurring has emphasized automatic choice of regularization parameters, but no method is completely reliable. The significance of this work is in going back to basics: allowing the user, who has domain-specific knowledge, to guide the choice of regularization using statistical criteria rather than a bias guided by expectations about the image.

### 6.2.1 Key Contribution

- We developed a user-friendly MATLAB interface called **GIDE** for deblurring images that can be used by a novice user with little knowledge of the numerical methods.
- We included diagnostics that limit user bias by testing the statistically feasibility of the solution with the three residual diagnostics.

- We used **GIDE** in an undergraduate class as a educational tool teaching students about image deblurring.
- We included a variety of linear and nonlinear methods in **GIDE**: truncated SVD, Tikhonov, and total variation regularization for deblurring.

### 6.2.2 Future Work

- Use a more robust numerical implementation of the TV regularization, that is more reliable over all parameter selections. The split Bregman method is a possible algorithm. It has been shown to be efficient on large problems [31].
- Incorporate additional regularization methods.
- Develop **GIDE** into a more efficient stand alone computer application or possible web application for image deblurring that can handle larger images and more general PSF's.

## Chapter A: Duality

A *dual* problem is a companion problem to an optimization (*primal*) problem where the dual variable measures the sensitivity of the primal objective to changes in the primal constraints. The dual problem is useful to measure the sensitivity of the constraints and sometimes is computationally more convenient to solve.

Our overview of duality is based on [35, 57, 63]; please reference these texts for more details.

### A.1 Nonlinear Duality

A nonlinear optimization problem is given by

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{s.t. } c_i(\mathbf{x}) \leq 0 \text{ for } i = 1 \dots m. \end{aligned} \tag{A.1}$$

Suppose the problem has a solution  $\mathbf{x}^*$ .

The Lagrangian of (A.1) is given by

$$L(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_{i=1}^m \alpha_i c_i(\mathbf{x}), \tag{A.2}$$

where  $\boldsymbol{\alpha}$  is the Lagrange multiplier vector. Each element of the vector is the Lagrange multiplier for the corresponding constraint. Then the primal problem can

be expressed as

$$\min_{\mathbf{x}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{x}, \boldsymbol{\alpha}). \quad (\text{A.3})$$

If the objective and constraints are continuously differentiable at optimal solution  $\mathbf{x}^*$ , then the necessary conditions for optimality are given by the Karush-Kuhn-Tucker (KKT) condition.

**Definition 2** (Karush-Kuhn-Tucker). *The Necessary Conditions, often called the Karush-Kuhn-Tucker (KKT) conditions, are satisfied if at the point  $\mathbf{x}^*$  there exists an  $\boldsymbol{\alpha}^*$  such that*

- $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\alpha}^*) = 0$  (*stationary condition*),
- $c_i(\mathbf{x}^*) \leq 0 \ \forall i$  (*primal feasibility*),
- $\boldsymbol{\alpha}^* \geq 0$  (*dual feasibility*),
- $\alpha_i^* c(\mathbf{x}^*) = 0 \ \forall i$  (*complementary slackness condition*).

Notice that the complementary slackness condition implies that either the primal constraint is active or the associated Lagrange multiplier is zero at the solution, hinting to a relationship between the dual and primal problems. If the objective and inequality functions are convex and  $(\mathbf{x}^*, \boldsymbol{\alpha}^*)$  satisfies the KKT conditions then it is an optimal solution.

Now we will introduce the dual problem in the form of the Lagrangian dual. We can express the dual of (A.3) as the max-min of the min-max problem [35]. Then the dual problem is given by

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}). \quad (\text{A.4})$$

We see that the value of (A.4) is always bounded above by (A.3):

$$\min_x \max_{\alpha \geq 0} L(\mathbf{x}, \alpha) \geq \max_{\alpha \geq 0} \min_x L(\mathbf{x}, \alpha). \quad (\text{A.5})$$

We have strong duality when there exists  $(\mathbf{x}^*, \alpha^*)$  such that

$$L(\mathbf{x}^*, \alpha^*) = \max_{\alpha \geq 0} \min_x L(\mathbf{x}, \alpha) = \min_x \max_{\alpha \geq 0} L(\mathbf{x}, \alpha). \quad (\text{A.6})$$

This holds when the KKT conditions are satisfied and the problem is convex. In such cases, we can choose to solve either the primal problem or the dual problem, or to consider both problems together.

## A.2 Duality for SVM

The soft-margin SVM optimization problem, which we will call the primal problem, is given by

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \|\xi\|_1 \quad \text{s.t.} \quad u_i(\langle \mathbf{w}, \mathbf{p}_i \rangle - b) \geq 1 - \xi_i, \\ \xi_i \geq 0 \text{ for all } i = 1, \dots, k. \end{aligned} \quad (\text{A.7})$$

The Lagrangian of the soft margin SVM is

$$L(\mathbf{w}, b, \xi, \alpha, \mu) = \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^k \xi_i - \sum_{i=1}^k \alpha_i (u_i(\langle \mathbf{w}, \mathbf{p}_i \rangle - b) - (1 - \xi_i)) - \sum_{i=1}^k \mu_i \xi_i. \quad (\text{A.8})$$

The KKT stationary condition of (A.7) is given by

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \xi, \alpha, \mu) = \mathbf{w} - \sum_{i=1}^k \alpha_i u_i \mathbf{p}_i = \mathbf{0}, \quad (\text{A.9})$$

$$\nabla_b L(\mathbf{w}, b, \xi, \alpha, \mu) = \sum_{i=1}^k \alpha_i u_i = 0, \quad (\text{A.10})$$

$$\nabla_{\xi} L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu}) = C\mathbf{e} - \boldsymbol{\alpha} + \boldsymbol{\mu} = \mathbf{0}, \quad (\text{A.11})$$

where  $\mathbf{e}$  is a vector of ones. In this case the KKT conditions are sufficient.

The dual problem of (A.7) is given by

$$\max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j u_i u_j \langle \mathbf{p}_i, \mathbf{p}_j \rangle \quad (\text{A.12})$$

$$\text{s.t. } \sum_i u_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C. \quad (\text{A.13})$$

It can be verified that the KKT conditions for the dual are the same as the KKT conditions for the primal problem, so we can substitute the dual whenever it is more convenient.

### A.3 Duality for TV

The primal total variation (TV) problem can be expressed in min-max form as

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \max_{\|\mathbf{y}_i\| \leq 1} \mathbf{x}^T \mathbf{Dy}. \quad (\text{A.14})$$

Then the dual TV is

$$\max_{\|\mathbf{y}_i\| \leq 1} \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x}^T \mathbf{Dy}. \quad (\text{A.15})$$

In this case the function  $\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x}^T \mathbf{Dy}$  is convex in  $\mathbf{x}$  and concave (actually linear) and bounded for  $\|\mathbf{y}_i\| \leq 1$  and we have strong duality:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \max_{\|\mathbf{y}_i\| \leq 1} \mathbf{x}^T \mathbf{Dy} = \max_{\|\mathbf{y}_i\| \leq 1} \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \mathbf{x}^T \mathbf{Dy}. \quad (\text{A.16})$$

Notice that

$$\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^m \mathbf{y}_i^T \mathbf{D}_i \mathbf{x} = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \gamma \sum_{i=1}^m \|\mathbf{D}_i^T \mathbf{x}\|. \quad (\text{A.17})$$



Equality holds when  $\mathbf{y}_i^T \mathbf{D}_i^T \mathbf{x} = \|\mathbf{D}_i^T \mathbf{x}\|$  for every  $i$  where  $\mathbf{D}_i^T \mathbf{x} \neq 0$  and  $\|\mathbf{y}_i\| \leq 1$ .

Then solving the following equations gives us the optimal solution to the primal and the dual problem:

$$\mathbf{A}^T \mathbf{A} \mathbf{x} - \mathbf{b} + \gamma \mathbf{D} \mathbf{y} = 0, \quad (\text{A.18})$$

$$\|\mathbf{D}_i^T \mathbf{x}\| \mathbf{y}_i - \mathbf{D}_i^T \mathbf{x} = 0, \quad (\text{A.19})$$

and

$$\|\mathbf{y}_i\| \leq 1. \quad (\text{A.20})$$

## Chapter B: Level Set Methods

Consider the two-dimensional problem where the initial curve is a circle with radius  $\sqrt{L}$  centered at the origin. We can define the curve in three dimensions as the  $L$  level set of

$$\Gamma_L = \{(x, y) \mid \psi_0(x, y) = L\}. \quad (\text{B.1})$$

where

$$\psi_0(x, y) = x^2 + y^2 = L. \quad (\text{B.2})$$

In general the model of the evolution of any curve, as it grows at a velocity  $\mathbf{v}$ , is given by the Cauchy problem

$$\psi_t + \mathbf{v} \cdot \nabla \psi = 0, \psi(x, y, 0) = \psi_0(x, y), \quad (\text{B.3})$$

or

$$\psi_t + F \|\nabla \psi\| = 0, \psi(x, y, 0) = \psi_0(x, y), \quad (\text{B.4})$$

where  $F = \mathbf{v} \cdot \mathbf{n}$  and  $\mathbf{n} = \frac{\nabla \psi}{\|\nabla \psi\|}$ .

We will first consider the case when  $F$  is constant. Without loss of generality, consider the initial curve where  $L = 0$  (point). In this case the solution to the PDE is a cone with vertex  $(0, 0, 0)$ . In general  $F$  will be a function of the position, other geometrical quantities, and/or, physical quantities related to the application

[84].

## B.1 Level Set Methods for Boundary Detection

In using level set methods for boundary detection for images we want to evolve an initial curve over a topology dependent on the image  $\mathbf{I}$ . There are two commonly cited level set methods for image boundary detection and segmentation.

The first method introduced by [10] uses a function  $F$  of the curvature ( $k$ ) and an edge indicator function ( $g$ ). Their level set method is given by

$$\psi_t - g(x, y) \|\nabla \psi\| (k + v) = 0, \psi(x, y, 0) = \psi_0(x, y). \quad (\text{B.5})$$

where  $v$  is a positive constant,

$$k = \text{div} \left( \frac{\nabla \psi}{\|\nabla \psi\|} \right), \quad (\text{B.6})$$

and

$$g(x, y) = \frac{1}{1 + \|\nabla \mathbf{I}(x, y)_\sigma\|^2}, \quad (\text{B.7})$$

where  $\mathbf{I}_\sigma$  is an image that has been convolved with the Gaussian filter (smoothed).

We see that the evolution of the curve is controlled by the gradient map of the image and the curvature of the evolving curve. For example for large gradients, where there is likely to be an edge,  $g(x, y)$  approaches zero and the curve evolution slows to zero. For large curvature, the velocity will increase resulting in a more smoothed curve.

The second, called the Chan-Vese algorithm [13], is related to the Mumford-Shah algorithm [55] which approximates an image by a piecewise smooth function.

The Chan-Vese algorithm assumes that the image can be partitioned into background and foreground where there exists a curve that separates the regions. Assume that the image  $\mathbf{I}$  can be approximated by a piecewise constant image of two distinct values. We can assign  $c_1$  and  $c_2$  as constants given by the average of  $\mathbf{I}$  within some curve and outside the curve respectively. Then to find the separating curve we want to minimize the energy function

$$E(\psi, c_1, c_2; \mathbf{I}_0) = \mu \int_{\Omega} \delta(\psi) \|\nabla \psi\| dx dy + \lambda_1 \int_{\Omega} |\mathbf{I}_0 - c_1|^2 H(\psi) dx dy + \lambda_2 \int_{\Omega} |\mathbf{I}_0 - c_2|^2 (1 - H(\psi)) dx dy \quad (\text{B.8})$$

where  $H$  is the Heaviside function given by  $H(z) = 1$  if  $z \leq 0$  and  $H(z) = 0$  otherwise and the Dirac measure  $\delta = \frac{d}{dz} H(z)$ . The first term is a regularization term controlling the length of the curve. The curve that minimizes (B.8) is the curve that best separates the image into background and foreground. To write it as a level set method we first need to find the Euler-Lagrange equation where it gives the governing PDE of the evolution of  $\psi$ . To solve the PDE numerically, the Heaviside function needs to be replaced with a continuously differentiable approximation of  $H$  [13] suggesting the continuously differentiable equation  $H_{\epsilon}(z) = \frac{1}{2}(1 + \frac{1}{\pi} \arctan(\frac{z}{\epsilon}))$ .

## Bibliography

- [1] Cancer Facts & Figures, American Cancer Society, 2013. <http://www.cancer.org/acs/groups/content/@epidemiologysurveillance/documents/document/acspc-036845.pdf>.
- [2] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroup: Theory of Positive Definite and Related Functions*. Springer, New York, N.Y., 1984.
- [3] Fred W Billmeyer and Max Saltzman. *Principles of Color Technology*. Wiley & Sons, New York, N.Y., 1981.
- [4] Sabri Boughorbel, J-P Tarel, and Nozha Boujemaa. Conditionally positive definite kernels for SVM based image recognition. In *Proceedings of the IEEE International Conference Multimedia and Expo*, pages 113–116. IEEE, 2005.
- [5] Wlodzimierz Bryc. Rotation invariant distributions. In *The Normal Distribution*, volume 100 of *Lecture Notes in Statistics*, pages 51–69. Springer, New York, N.Y., 1995.

- [6] Sandy Campbell. Digital imaging in dermatology, December 2005. <https://www.dermquest.com/professional-education/practice-management/digital-imaging-in-dermatology/>.
- [7] Gustavo Camps-Valls, Luis Gomez-Chova, Jordi Muñoz-Marí, Joan Vila-Francés, and Javier Calpe-Maravilla. Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 3(1):93–97, 2006.
- [8] Jonathan C Carr, Richard K Beatson, Bruce C McCallum, W Richard Fright, Tim J McLennan, and Tim J Mitchell. Smooth surface reconstruction from noisy range data. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pages 119–ff. ACM, 2003.
- [9] Thierry Carron and Patrick Lambert. Color edge detector using jointly hue, saturation and intensity. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 977–981. IEEE, 1994.
- [10] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.
- [11] Tony F. Chan, Gene H. Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *Lecture Notes in Control and Information Sciences*, 219:241–251, 1996.
- [12] Tony F. Chan and Jianhong Shen. *Image Processing and Analysis*. SIAM, Philadelphia, P.A., 2005.
- [13] Tony F Chan and Luminita A Vese. Active contours without edges.

- IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
- [14] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Network*, 10(5):1055–1064, 1999.
  - [15] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.
  - [16] Heng-Da Cheng, XH Jiang, Ying Sun, and Jingli Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.
  - [17] Vladimir Cherkassky and Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1):113–126, 2004.
  - [18] Colorimetrix. <http://www.colorimetrix.com/>.
  - [19] Christine Connolly and T Fleiss. A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. *IEEE Transactions on Image Processing*, 6(7):1046–1048, 1997.
  - [20] Dermoscopy Atlas. <http://www.dermoscopyatlas.com/>.
  - [21] Ilze Diebele, Ilona Kuzmina, Alexey Lihachev, Janis Kapostinsh, Alexander Derjabo, Lauma Valeine, and Janis Spigulis. Clinical evaluation of melanomas and common nevi by spectral imaging. *Biomedical Optics Express*, 3(3):467–472, 2012.
  - [22] Salih Dikbas, T Arid, and Yucel Altunbasak. Chrominance edge preserving grayscale transformation with approximate first principal component for color

- edge detection. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages II–261. IEEE, 2007.
- [23] Soumya Dutta and Bidyut B Chaudhuri. A color edge detection algorithm in RGB color space. In *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing*, pages 337–340. IEEE, 2009.
- [24] Easymeasure - Measure with your Camera! <http://www.caramba-apps.com/easymeasure/>.
- [25] Timo Eckhard, Eva M. Valero, Javier Hernández-Andrés, and Ville Heikkinen. Evaluating logarithmic kernel for spectral reflectance estimation—effects on model parametrization, training set size, and number of sensor spectral channels. *Journal of the Optical Society of America*, 31(3):541–549, March 2014.
- [26] Stefan Fischer, Philippe Schmid, and Joël Guillod. Analysis of skin lesions with pigmented networks. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 323–326. IEEE, 1996.
- [27] Robert Fisher and Jonathan Rees. DERMOFIT: A cognitive prosthesis to aid focal skin lesion diagnosis. <http://homepages.inf.ed.ac.uk/rbf/DERMOFIT/>.
- [28] Wayne A. Fuller. *Introduction to Statistical Time Series*. Wiley-Interscience, New York, N.Y., 1996.
- [29] P Ganesan, V Rajini, and R Immanuvel Rajkumar. Segmentation and edge detection of color images using CIELAB color space and edge detectors. In *Proceedings of the International Conference on Emerging Trends in Robotics*



- and Communication Technologies*, pages 393–397. IEEE, 2010.
- [30] Harald Ganster, Axel Pinz, R Rohrer, Ernst Wildling, Michael Binder, and Harald Kittler. Automated melanoma recognition. *IEEE Transactions on Medical Imaging*, 20(3):233–239, 2001.
  - [31] Tom Goldstein and Stanley Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
  - [32] David Delgado Gómez, Constantine Butakoff, Bjarne Kjaer Ersboll, and William Stoecker. Independent histogram pursuit for segmentation of skin lesions. *IEEE Transactions on Biomedical Engineering*, 55(1):157–161, 2008.
  - [33] H Gomez-Moreno, P Gil-Jimenez, S Lafuente-Arroyo, R Vicen-Bueno, and R Sanchez-Montero. Color images segmentation using the support vector machines. *Recent Advances in Intelligent Systems and Signal Processing*, pages 151–155, 2003.
  - [34] Amy A Gooch, Sven C Olsen, Jack Tumblin, and Bruce Gooch. Color2gray: Saliency-preserving color removal. In *ACM Transactions on Graphics*, volume 24, pages 634–639. ACM, 2005.
  - [35] Igor Griva, Stephen G Nash, and Ariela Sofer. *Linear and Nonlinear Optimization*. SIAM, Philadelphia, P.A., 2009.
  - [36] Handyscope. <http://www.handyscope.net/>.
  - [37] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Numerical Aspects of Linear Inversion. SIAM, Philadelphia, P.A., 1998.
  - [38] Per Christian Hansen. *Discrete Inverse Problems: Insight and Algorithms*, volume 7. SIAM, Philadelphia, P.A., 2010.

- [39] Per Christian Hansen, James G Nagy, and Dianne P O’leary. *Deblurring Images: Matrices, Spectra, and Filtering*, volume 3. SIAM, Philadelphia, P.A., 2006.
- [40] Arne-Jens Hempel and Steffen F Bocklisch. Fuzzy classification of nonconvex data-inherent structures. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Methods*, pages 416–425. Springer, New York, N.Y., 2010.
- [41] Marylesa Marie Howard. *Computational Methods for Support Vector Machine Classification and Large-Scale Kalman Filtering*. PhD thesis, University of Montana, 2013.
- [42] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.
- [43] Jin Hyuk Jung. *Adaptive Constraint Reduction for Convex Quadratic Programming and Training Support Vector Machines*. PhD thesis, University of Maryland, 2008.
- [44] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of the European Conference on Computer Vision*, pages 502–516. Springer, 2012.
- [45] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

- [46] Leafsnap. <http://leafsnap.com/>.
- [47] Peng Liu and Dingsheng Liu. Selection of regularization parameter based on generalized cross-validation in total variation remote sensing image restoration. In *Remote Sensing*, pages 78301P–78301P. International Society for Optics and Photonics, 2010.
- [48] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.
- [49] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.
- [50] John Mashford, Mike Rahilly, Paul Davis, and Stewart Burn. A morphological approach to pipe image interpretation based on segmentation by support vector machine. *Automation in Construction*, 19(7):875–883, 2010.
- [51] Ammara Masood and Adel Ali Al-Jumaily. Computer aided diagnostic support system for skin cancer: A review of techniques and algorithms. *International Journal of Biomedical Imaging*, 2013:22, 2013.
- [52] Melafind. <http://www.melafind.com/>.
- [53] Ajay Mittal, Sanjeev Sofat, and Edwin Hancock. Detection of edges in color images: A review and evaluative comparison of state-of-the-art techniques. *Autonomous and Intelligent Systems*, pages 250–259, 2012.

- [54] Molemax. [http://www.dermamedicalsystems.com/index.php?menu\\_id=115](http://www.dermamedicalsystems.com/index.php?menu_id=115).
- [55] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [56] James G. Nagy. Restoretool, 2004. <http://www.mathcs.emory.edu/~nagy/RestoreTools/>.
- [57] Stephen G. Nash. *Linear and Nonlinear Programming*. McGraw-Hill, New York, N.Y., 1996.
- [58] Remakant Nevatia. A color edge detector and its use in scene segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 7(11):820–826, 1977.
- [59] Ehsan Nezhadarya and Rabab Kreidieh Ward. A new scheme for robust gradient vector estimation in color images. *IEEE Transactions on Image Processing*, 20(8):2211–2220, 2011.
- [60] Mark Nischik and Clemens Forster. Analysis of skin erythema using true-color images. *IEEE Transactions on Medical Imaging*, 16(6):711–716, 1997.
- [61] CL Novak and SA Shafer. Color edge detection. *Proceedings of DARPA Image Understanding Workshop*, 1:35–37, 1987.
- [62] Hiroshi Ohshima, Hirotugu Takiwaki, Kumiko Washizaki, Akira Ishiko, Masatoshi Itoh, and Hiromi Kanto. Quantitative evaluation of patch test reactions: A comparison between visual grading and erythema index image analysis. *Skin Research and Technology*, 17(2):220–225, 2011.
- [63] Dianne O’Leary. *Scientific Computing with Case Studies*. SIAM, Philadelphia,

- phia, P.A., 2009.
- [64] Ilse C.F. Ipsen Pierre Gremaud and Ralph Smith. 2012 Industrial mathematical statistical modeling workshop for graduate students technical report. Technical report, Center for Research in Scientific Computation, North Carolina State University, 2012.
  - [65] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.
  - [66] Sadaf Qazi, Karen Panetta, and Sos Agaian. Detection and comparison of color edges via median vased PCA. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 702–706. IEEE, 2008.
  - [67] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the 24th International Conference on Machine learning*, pages 775–782. ACM, 2007.
  - [68] Darrell S Rigel, Julie Russak, and Robert Friedman. The evolution of melanoma diagnosis: 25 years beyond the ABCDs. *CA: A Cancer Journal for Clinicians*, 60(5):301–316, 2010.
  - [69] Wendy E Roberts. Skin type classification systems old and new. *Dermatologic Clinics*, 27(4):529–533, 2009.
  - [70] Sheldon Ross. *A First Course in Probability, Sixth Edition*. Prentice Hall, Upper Saddle River, N.J., 2002.
  - [71] Stefan Rüping. A simple method for estimating conditional probabilities for SVMs. Technical report, Universität Dortmund, 2004.

- [72] Bert W. Rust. Parameter selection for constrained solutions to ill-posed problems. *Computing Science and Statistics*, 32:333–347, 2000.
- [73] Bert W. Rust and Dianne O’Leary. Residual programs for choosing regularization parameters for ill-posed problems. *Inverse Problems, Invited Paper*, 24(034005):30, 2008.
- [74] Mark A Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *IEEE Computer Vision and Pattern Recognition*, volume 2, page 166. IEEE, 1999.
- [75] Bernhard Scholkopf. The kernel trick for distances. *Advances in Neural Information Processing Systems*, pages 301–307, 2001.
- [76] Bernhard Scholkopf and Alex Smola. *Learning with Kernels*. The MIT Press, Cambridge, M.A., 2002.
- [77] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, M.A., 2004.
- [78] Margarida Silveira and Jorge S Marques. Level set segmentation of dermoscopy images. In *IEEE International Symposium Biomedical Imaging*, pages 173–176, 2008.
- [79] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.
- [80] Cesar R Souza. Kernel functions for machine learning applications, 2010. <http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html>.

- [81] H Takiwaki, S Shirai, Y Kanno, Y Watanabe, and S Arase. Quantification of erythema and pigmentation using a videomicroscope and a computer. *British Journal of Dermatology*, 131(1):85–92, 2006.
- [82] PE Trahanias and AN Venetsanopoulos. Color edge detection using vector order statistics. *IEEE Transactions on Image Processing*, 2(2):259–264, 1993.
- [83] PE Trahanias and AN Venetsanopoulos. Vector order statistics operators as color edge detectors. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):135–143, 1996.
- [84] Richard Tsai and Stanley Osher. Level set methods and their applications in image science. *Communications in Mathematical Sciences*, 1(4):1–20, 2003.
- [85] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Network*, 10(5):988–999, 1999.
- [86] Rosaly Vieira dos Santos, Richard G Titus, and Hermênio Cavalcante Lima. Objective evaluation of skin prick test reactions using digital photography. *Skin Research and Technology*, 13(2):148–153, 2007.
- [87] Curtis R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, P.A., 2002.
- [88] Curtis R. Vogel and M. Oman. Iterative methods for total variation denoising. *SIAM Journal Scientific Computing*, 17:227–238, 1996.
- [89] Stefan Wöhrle, Kornelia Vigl, Michael Binder, Georg Stingl, and Michael Prinz. Automated measurement of skin prick tests: An advance towards exact calculation of wheal size. *Experimental Dermatology*, 15(2):119–124, 2006.
- [90] Yue Wu. Chan-ese active contours without edges, 2009.

<http://www.mathworks.com/matlabcentral/fileexchange/23445-chan-veese-active-contours-without-edges>.

- [91] Tadamasa Yamamoto, Hirotsugu Takiwaki, Seiji Arase, and Hiroshi Ohshima. Derivation and clinical application of special imaging by means of digital cameras and IMAGE J FREWARE FOR QUANTIFICATION OF ERYTHEMA AND PIGMENTATION. *Skin Research and Technology*, 14(1):26–34, 2008.
- [92] JIE YANG, WEIER LU, AND ALEX WAIBEL. *Skin-Color Modeling and Adaptation*. SPRINGER, NEW YORK, N.Y., 1997.
- [93] ALI K YETISEN, JL MARTINEZ-HURTADO, ANGEL GARCIA-MELENDEZ, FERNANDO DA CRUZ VASCONCELLOS, AND CHRISTOPHER R LOWE. A SMARTPHONE ALGORITHM WITH INTER-PHONE REPEATABILITY FOR THE ANALYSIS OF COLORIMETRIC TESTS. *Sensors and Actuators B: Chemical*, 196:156–160, 2014.